A Novel Inpainting Framework for Virtual View Synthesis

Thesis

A Novel Inpainting Framework for Virtual View Synthesis

Smarti Reel

B.Tech, M.Tech

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Department of Computing and Communications

Faculty of Mathematics, Computing and Technology

The Open University

Milton Keynes, United Kingdom

Submitted on 30^{th} June 2016

I dedicate this work to ...

My Parents, whose innumerable sacrifices are the reason I could achieve this milestone.

My Husband, who has been my indispensable source of love, inspiration and encouragement throughout this journey and the journey of life.

You are my ROCK.

My Sister, for her unconditional love, trust and faith in me.

Acknowledgements

The PhD journey has been an interesting and intensive experience for me. This note of thanks is the finishing touch on my thesis and I would like to reflect on the people who have supported and inspired me throughout this period to achieve this milestone.

First and foremost, I would like to express sincere gratitude to my supervisors Dr. Patrick Wong, Prof Laurence S. Dooley and Dr. Gene Cheung for encouraging my research and allowing me to grow as an independent researcher. Thanks to Dr. Patrick Wong for his excellent guidance, endless support and most of all, his kindness. He has always been approachable and immensely understanding during this journey.

Prof. Laurence S Dooley has been a tremendous mentor for me. His immense experience, unfailing advice has greatly benefited me in understanding the research protocol. His philosophy of '*golden thread*' helped me weave together various strands of this thesis. I am indebted to Dr. Gene Cheung for having given me the opportunity to work under his expert supervision during my internship at NII. His in-depth knowledge and insightful comments have incented me to widen my research from various perspectives. Without the guidance and constant motivation of all my supervisors, this work would not have been possible.

I would like to thank my third party monitor Dr. Soraya Kouadri Mostéfaoui who has been my moral booster during overwhelming times. I also wish to thank Dr. Adrian Poulton and Dr. Helen Donelan for their valuable advices and constructive feedback during probation viva.

I am grateful to all the past and current members of *XGMT Research Group* and my fellow colleagues in *S1005* for their constant support and valuable discussions which has deeply enriched my research experience. A special thanks to Dimitris for proof-reading the final version of my thesis chapters. I would also like to thank to my dear friends Inga, Anko, Julia, Tejaswini, Advait, Linda and Thomas for their unconditional support and care during various highs and lows of this journey. I greatly value their friendship and deeply appreciate their belief in me.

Thanks to all the administrative staff in the department and the research school with special appreciation for Donna Deacon, Su Prior and Linda. Last but not the least, a special thanks to IT department for excellently maintaining my machine and the security staff for providing me a secure workplace during numerous long weekends of work spent in the university.

Abstract

Multi-view imaging has stimulated significant research to enhance the user experience of free viewpoint video, allowing interactive navigation between views and the freedom to select a desired view to watch. This usually involves transmitting both textural and depth information captured from different viewpoints to the receiver, to enable the synthesis of an arbitrary view. In rendering these *virtual views*, perceptual holes can appear due to certain regions, hidden in the original view by a closer object, becoming visible in the virtual view. To provide a high quality experience these holes must be filled in a visually plausible way, in a process known as *inpainting*. This is challenging because the missing information is generally unknown and the hole-regions can be large. Recently depth-based inpainting techniques have been proposed to address this challenge and while these generally perform better than non-depth assisted methods, they are not very robust and can produce perceptual artefacts.

This thesis presents a new inpainting framework that innovatively exploits depth and textural self-similarity characteristics to construct subjectively enhanced virtual viewpoints. The framework makes three significant contributions to the field: i) the exploitation of view information to *jointly inpaint textural and depth hole regions*; ii) the introduction of the novel concept of *self-similarity characterisation* which is combined with relevant depth information; and iii) an *advanced self-similarity characterising scheme* that automatically determines key spatial transform parameters for effective and flexible inpainting.

The presented inpainting framework has been critically analysed and shown to provide superior performance both perceptually and numerically compared to existing techniques, especially in terms of lower visual artefacts. It provides a flexible robust framework to develop new inpainting strategies for the next generation of interactive multi-view technologies.

Declaration

The work presented in this thesis is an original contribution of the author. Parts of this thesis have appeared in the following:

Peer-Reviewed Publications:

1. **Reel, S.**, Cheung, G., Wong, P., and Dooley, L. S., (2013). Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis. In *IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (*APSIPA 2013*), pages 1-7, Kaohsiung, Taiwan.
   doi : 10.1109/APSIPA.2013.6694249

2. **Reel, S.**, Wong, K. C. P., Dooley, L. S., and Cheung, G., (2014). Disocclusion hole-filling in DIBR-synthesized images using multi-scale template matching. In *IEEE Visual Communications and Image Processing Conference* (*VCIP 2014*), pages 494-497, Valletta, Malta.
   doi : 10.1109/VCIP.2014.7051614

Poster Presentations:

1. **Reel, S.**,(2014). 3D Disocclusion Hole Inpainting. In *IET Midlands Power Network Engineering Event (in association with industry)*, Coventry University, UK, Mar 18, 2014.

2. **Reel, S.**, (2014). Virtual View Inpainting. In *Life Beyond the PhD Conference 2014*, Cumberland Lodge, Windsor Great Park, UK, Aug 26, 2014.

3. **Reel, S.**,(2015). 3D Multiview Image Inpainting. In *BCS London Hopper Colloquium 2015)*, BCS Headquarters, London, UK, May 20, 2015.

Research Awards:

1. *Student Travel Award* by IEEE Circuits & System Society, U.S.A. (to attend VCIP'2014 Conference in December 2014).

2. *London Hopper Colloquium 2015 Finalist Prize* for Research Spotlight Competition sponsored by BCS Academy and University College London, at BCS Headquarters, London, UK, May 20, 2015.

Smarti Reel

Submitted on 30^{th} June 2016

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
3D-TV	Three Dimensional Television
ASC	Advanced Self-similarity Characterisation
ASC-JTDI	ASC based Joint Texture-Depth Inpainting
BCP	Boundary Candidate Patch
BG	Background
BP	Boundary Patch
BTP	Boundary Target Patch
CP	Candidate Patch
DAI	Depth-Assisted Inpainting
DIBR	Depth Image-Based Rendering
DS-DIBR	Double Sided-DIBR
EBI	Exemplar-Based Inpainting
FFT	Fast Fourier Transform
FG	Foreground
FMT	Fourier Mellin Transform
FT	Fourier Transform
FTV	Free-viewpoint TV
FVV	Free-Viewpoint Video
JTDI	Joint Texture-Depth Inpainting

LPT	Log Polar Transform
LUT	Look-Up Table
LUT-SR	Look-Up Table-SR
MRF	Markov Random Field
MIT	Massachusetts Institute of Technology
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MVD	Multiview Video-plus-Depth
MVSV	Multiview View Synthesis
NaN	Not a Number
NSW	Neighbourhood Search Window
PC	Personal Computer
PDE	Partial Differential Equations
PSNR	Peak Signal-to-Noise Ratio
RAM	Random Access Memory
RGB	Red Green Blue
SC	Self-similarity Characterisation
SC-JTDI	Self-similarity Characterisation based JTDI
SI	Supplementary Information
SP	Scale Parameters
SR	Scale Rotation
SSE	Sum of Squared differences
SS-DIBR	Single Sided-DIBR
SSIM	Structural SIMilarity

TM	Template Matching
TOF	Time-of-Flight
TP	Target Patch
TV	Television

List of Symbols

α	Virtual camera position
β	Scale value
γ	Normalisation factor
ξ	Rotation parameter
ρ	Radius
θ	Angle
τ_d	Minimum distance threshold
τ_e	Luminance threshold
Φ	Source region
Ω	Hole region
$\delta\Omega$	Hole region boundary
$\Psi_{\hat{q}}$	Candidate Patch
$\Psi_{\hat{p}}$	Target patch
Ψ_p	Square template used for priority computation
$\nabla I_p^{\perp}$	Isophote
$\forall$	For all
a	Scale value
Ac	Columns of input image
Ar	Rows of input image
$A_i(x, y)$	Pixel patch in Cartesian coordinates

B	Baseline
c	Correlation coefficient
C_i	Camera at i^{th} position.
$C(p)$	Candidate term
$C(\hat{p})$	Updated confidence values
CP^i	i^{th} candidate patch
CP^i_β	i^{th} candidate patch for given β
$C_{virtual}$	Virtual camera viewpoint
D	Depth cut-off values
d	Disparity
d_{ij}	Distance between i and j
F	Focal length
f_i	Luminance feature vector
$\overrightarrow{g}_i$	1D descriptor
$\overrightarrow{g}'_i$	1D descriptor for scaled and rotated patch
$\overrightarrow{G}_i$	Fourier magnitude of $\overrightarrow{g}_i$
$\overrightarrow{G}'_i$	Fourier magnitude of $\overrightarrow{g}'_i$
h	Number of candidates patches in X
I	Input image
I_T	Texture image
I_D	Depth image
l	Patch size for neighbourhood search window
$L(p)$	Depth variance term
n_p	Orthogonal unit vector

N	Number of pixels in frame
$o_{x,v}$	Principle point offset for virtual view
$o_{x,r}$	Principle point offset for reference view
p	Centre pixel of square template
$\hat{p}$	Centre pixel of target patch
p_k	Probability
$P(p)$	Priority term
r	Log radius
R_{peak}	Maximum magnitude peak
s	Total number of segments
S_{β}^{i}	i^{th} patch for given β in search space
TP^{i}	i^{th} target patch
$t_{x,v}$	Translational vector for virtual view
$t_{x,r}$	Translational vector for reference view
u_i	i^{th} texture segment
U_b	Selected background segment
U	All texture segments
V_T	Virtual texture image
V_D	Virtual depth image
X	Candidate search space
x_i	Reference pixel value at location i
$\widetilde{x}_i$	Pixel value to measure distortion against x_i
x_j	j^{th} candidate patch in X
x_{max}	Peak pixel value

z_i	Depth cut-off values for i^{th} segment
Z	Depth
Z_{far}	Farther from camera
Z_{near}	Closer to camera
Z_p	Target depth patch
$\overline{Z}_p$	Target depth mean
$Z_{\hat{p}}$	Target depth pixel
$Z_p(q)$	Depth value at the pixel location q
Z_q	Candidate depth patch
$\overline{Z}_q$	Candidate depth mean
$Z_{\hat{q}}$	Candidate depth pixel

Chapter 1

Introduction

1.1 Overview

The invention of television (TV) has revolutionised the world through visual and audio technology. The British Broadcasting Corporation debuted the world's first regular TV service in 1936, and since then, there have been much significant advances in the field of video technology from picture capture to displays. TV broadcasting has grown from the passive broadcasts to providing interactive on-demand type services (Owens, 2016; Zhu et al., 2012). The demand for visual media has led to ever-growing research which has resulted in today's immersive experience of Ultra High Definition TV and three-Dimensional (3D) TV (Kryszkiewicz et al., 2015; Kubota et al., 2007).

3D-Video gives the user an experience as if they are watching real-world objects through a window rather than looking at images projected onto a flat panel.

Typically, 3D-Video is obtained from a set of synchronised cameras by capturing the same scene from two different viewpoints (Kondoz and Dagiuklas, 2014). This has led to the development of multi-view technology which enables the capture of different views of the same scene. A 3D multi-view capturing format facilitates new applications like Free-Viewpoint Video (FVV), which allow the free navigation between viewpoints for a seamless viewing experience (Emori et al., 2015; Tanimoto et al., 2011). The Moving Picture Experts Group (MPEG) started developing Free-viewpoint TV (FTV) in 2001, and plans to establish a new FTV framework to revolutionise the viewing experience, targeting particularly the 2020 Tokyo Olympics (Vito, 2015).

The standard 3DTV and FTV broadcasting chain representing *virtual view* rendering is shown in Figure 1.1 and comprises a 3D multi-view capturing unit (cameras shown as $C_1 .... C_n$) that acquires data from real-world scenes as textural (i.e. colour) images together with depth maps (Scharstein and Szeliski, 2003). The captured data is then aligned and efficiently encoded for transmission. At the receiver, the data is then decoded and the views rendered before final display. The view rendering process gives the viewer the freedom of navigating through the scene to choose different viewpoints, however this requires a very large number of cameras, so capturing and broadcasting arbitrary viewpoints for FVV incur excessively high coding overheads, expensive processors and high broadcasting costs. Hence, instead of employing large numbers of multi-view 3D cameras at the encoder, the alternative is to apply rendering techniques to synthesise intermediate views known as *virtual views* at decoder. This means a smaller number of cameras is then required to capture the scene information, so minimising the overall trans-

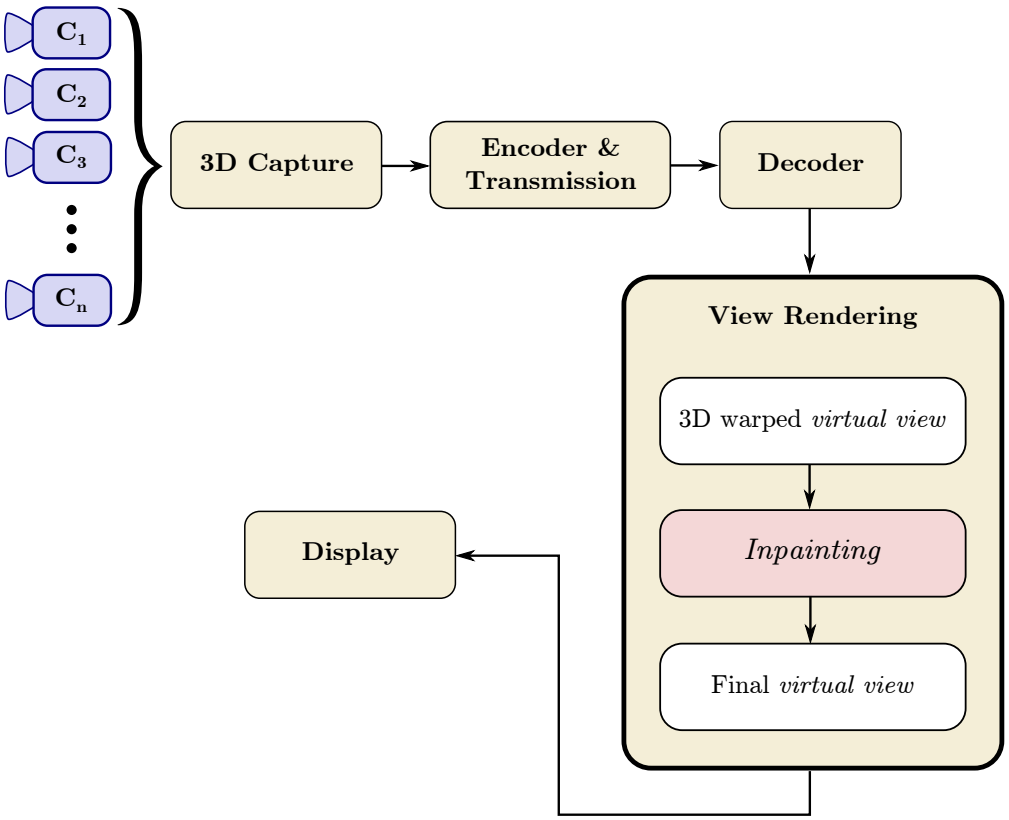


Figure 1.1: Block diagram of basic FVV system

mission cost (Kubota et al., 2007; Smolic and Kauff, 2005). At the receiver, the finite number of views can be used to render multiple intermediate views allowing the user to interactively navigate among various viewpoints and provide content for display. Thus, the view rendering process represents a significant step in FVV.

The most commonly used rendering method to synthesise virtual views is called *Depth Image-Based Rendering* (DIBR) (Muller et al., 2011; Tian et al., 2009; Vetro et al., 2008). To synthesise a virtual view using DIBR, 3D warping (Tian et al., 2009) is performed which essentially projects pixels in the camera-captured reference view to corresponding pixel locations in the new viewpoint (see Section 2.4.1 for details). However, the synthesised view often contains artefacts caused by missing information or so-called *holes* which degrade the perceptual experience.

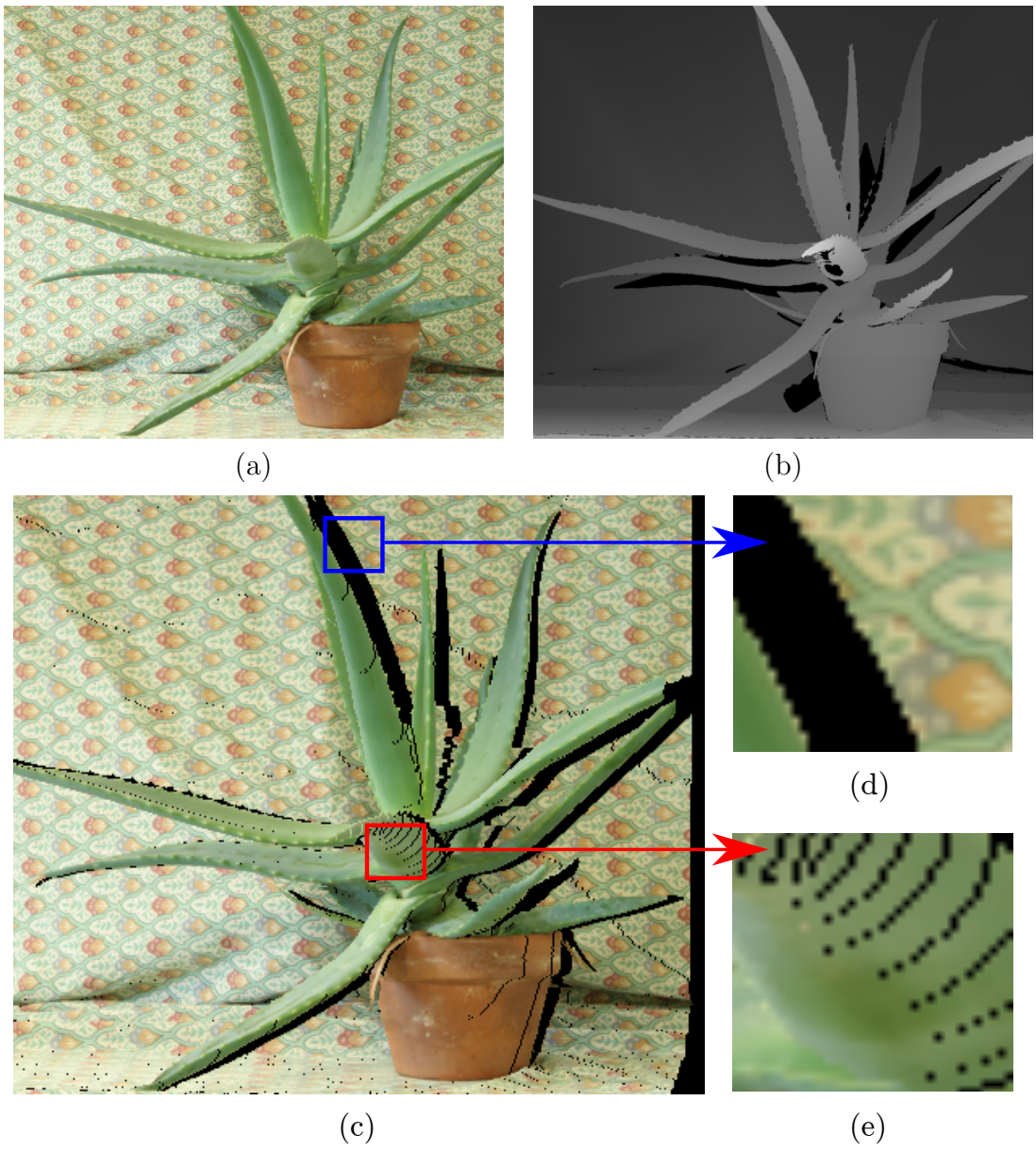


Figure 1.2: Reference views for: (a) texture; (b) depth; (c) 3D warped texture virtual view illustrating (d) disocclusion holes and (e) cracks.

For example, Figures 1.2 (a) and (b) show camera captured texture and depth reference view of the *Aloe* image (Scharstein and Pal, 2007), which is used to synthesise the virtual view located to the right hand side of the reference view (as explained in section 3.3). The plant is considered as a *foreground* (FG) object (i.e. it is closer to the camera) with respect to the patterned *background* (BG) area which is farther away. After 3D warping, due to the viewpoint change, the

resulting view contains missing pixels, shown as the *black* regions in Figure 1.2 (c). These missing pixels can be classified into two main categories (Zhu et al., 2012) : i) *cracks* and ii) *disocclusion holes*, with examples of each being illustrated in the zoomed-in regions displayed in Figures 1.2 (e) and (d) respectively.

Cracks appear when projected pixel position is rounded to the nearest integer and since they are generally one pixel in width (Muddala et al., 2013; Tian et al., 2009), they can be filled using traditional interpolation and conventional filtering techniques (Mori et al., 2008; Oh et al., 2009). However, disocclusions are spatial regions in the virtual view that were occluded by FG objects in the captured camera view(s) and which during viewpoint change, become exposed or disoccluded in the virtual view. A hole with no corresponding pixels in the reference view is thus known as a *disocclusion hole* (Cheung et al., 2015; Daribo and Pesquet-Popescu, 2010; Guillemot and Meur, 2014).

A major design objective is to suitably fill the disocclusion holes to achieve satisfactory perceptual quality. In the literature, appropriate strategies for filling missing pixels, are referred to as *inpainting* or *hole-filling* (Bertalmio et al., 2000; Bugeau et al., 2010; Buyssens et al., 2015) and this is the core process in synthesising a virtual view. The growing popularity of FVV has led to an increased interest in developing suitable inpainting techniques for DIBR-synthesised views. The next section discusses some of the more popular inpainting methods used to fill missing pixels in virtual views.

1.2 Inpainting of Synthesised Views

Inpainting is the process of reconstructing missing or deteriorated regions such as scratches and holes in an image in a perceptually undetectable manner. The 2D inpainting (i.e. in the absence of depth information) methods can be broadly classified as: 1) geometry-based techniques which use partial differential equations (PDE) (Chan et al., 2002; Masnou and Morel, 1998); 2) texture-based methods use template matching (Kwatra et al., 2003) 3) exemplar-based schemes (Buyssens et al., 2015; Criminisi et al., 2004; Martanez-Noriega et al., 2012). PDE-based schemes perform well in preserving image structures and geometry, but tend to degrade when the hole area is large. Texture-based inpainting methods use *template matching* (TM) to fill in missing pixels by copying a fixed-size pixel region from a known spatial area where there are no holes, to one where there are. While these generally work well for large region inpainting, they do not preserve the image structure. Exemplar-based inpainting in contrast, combines the advantages of both geometry and texture based methods to achieve inpainting of large missing regions, while preserving structure, with the most well-known exemplar-based algorithm for regular texture images being proposed by (Criminisi et al., 2004).

Due to their structure and texture preserving properties, DIBR-synthesised views have mainly focused on exemplar-based approaches. Disocclusion holes, which appear due to missing BG information, are required to be filled from information in the BG regions. However, 2D inpainting algorithms are inadequate for filling disocclusion holes in virtual views because they cannot differentiate between FG from BG regions. This means they can falsely propagate FG information to

BG regions resulting in visual artefacts. Since depth information is also available for the view synthesis process, it has been investigated for improving the inpainting of disocclusion holes. (Ahn and Kim, 2013; Daribo and Pesquet-Popescu, 2010; Gautier et al., 2011) are all exemplar-based methods that exploit the (Criminisi et al., 2004) algorithm, but in addition, introduce a depth constraint to the hole-filling process. While these provided improved inpainting performance compared with non-depth-assisted methods, they still produce visual artefacts which must be removed in order to produce the best visual quality of the synthesised view.

1.3 Research Motivation

Generating high quality virtual views is especially challenging as the baseline distance between the reference camera and selected virtual viewpoint increases, leading to bigger holes. Disocclusion holes possess certain characteristics which needs to be considered in developing any new inpainting strategy:

- The order of hole-filling is vital. Exemplar-based methods inpaint texture regions with structural information first followed by homogeneous regions. However, for disocclusion holes, it is important to formulate the order so hole-filling starts from the BG boundary rather than the FG. If the filling order starts from a FG region, it can lead to serious error propagation in subsequent steps.

- Disocclusion holes tend to occur at FG object boundaries and are typically located on the border between FG and BG. The missing region belongs

to the BG so it is important to inpaint the disocclusion holes using only BG information. Inpainting with FG information results in more visual inconsistencies and artefacts.

- In practice, the synthesised virtual view contains disocclusion holes in both the texture and depth maps. However, various depth-based inpainting methods assume the availability of a complete depth map (Daribo and Pesquet-Popescu, 2010; Gautier et al., 2011). This assumption is unreasonable and inpainting of the synthesised depth map also requires attention alongside texture inpainting. The potential to effectively inpaint both texture and depth maps is an important aim as both are correlated and needs to be filled consistently. Also a synthesised virtual view can then be used as a reference view to construct other virtual views.

- Exemplar-based methods primarily use TM for hole-filling. The template is a square region of pixels of fixed size which is called a *patch*. During matching, the image is used as a search-space to find the best patch in accordance with the template patch and copies this to the missing region. However, finding adequate patches with similar information within an image is problematic, so generating a richer search space of potential patches for TM may markedly impact upon the inpainting quality.

- The choice of the best patch size for filling disocclusion holes in different images is debatable, so suitable mechanisms to determine the most appropriate patch size for inpainting are worthy of further investigation.

Using these distinct disocclusion hole features, the challenge of how to effectively

improve the visual and numerical inpainting performance was the main motivation behind this research, namely how best to exploit *a priori* knowledge about disocclusion holes for inpainting. This led to the overarching thesis research question and related objectives, which are discussed in the next section.

1.4 Research Question and Objectives

From the above discussion, the following research question was framed:

How can inpainting achieve high-quality virtual view synthesis?

The aim is to provide a high-quality virtual view which eliminates holes and minimises visual artefacts in synthesised views by providing an effective and superior inpainting solution for disocclusion hole-filling. After a detailed review of the existing inpainting methods for hole-filling, an inpainting paradigm that exploits depth information allied with image self-similarity characteristics has been identified as a fertile area of investigation.

To address this overarching research question, a set of three objectives were framed:

1. *To develop and critically evaluate a new joint texture-depth inpainting technique.*

   Justification: To investigate a new inpainting approach which jointly inpaints texture and depth pixels in disoccluded regions. Existing solutions (Daribo and Pesquet-Popescu, 2010; Meur et al., 2011) use depth information to

propose BG to FG filling order with an underlying assumption that a high-quality depth map is available at the virtual view for hole-filling. However this assumption is not valid for practical DIBR systems because the virtual view can have disocclusion holes in both the texture and depth maps. Also the filling order may not always perform BG to FG hole-filling. While some existing methods process depth maps before texture inpainting (Oh et al., 2009; Ramachandran and Rupp, 2012), others perform only texture hole-filling and completely ignore the depth inpainting (Wang et al., 2015). Thus, exploring new inpainting techniques which perform BG to FG hole-filling by jointly exploiting textural and depth information offers the potential for more robust and improved inpainting.

2. *To investigate how image self-similarity characterisation allied with the depth information can enhance the quality of inpainting.*

   Justification: Exemplar-based inpainting methods exploit the self-similarity of natural images to fill missing pixels by identifying similar pixel patterns within the image. There are however two key drawbacks: i) the scarcity of self-similar patches due to the limited search space in an image; and ii) TM schemes involve an exhaustive search process which is computationally expensive and leads to higher inpainting times. In view rendering, the texture image tends to possess transformation properties due to possible depth variations in either the image or in image patterns. This provided the motivation to investigate new inpainting strategies which exploit image transformation characteristics as well as using depth information to both improve the search space and provide faster inpainting.

Figure 1.3: Key contributions of inpainting framework with JTDI as core block.

3. *To critically synthesise an advanced self-similarity characterisation technique for inpainting disocclusion holes.*

   Justification: A new mechanism is devised that extends the idea of self-similarity characterisation to broaden the search space by incorporating additional image characteristic. A superior approach is modelled for the simultaneous determination of self-similarity characteristics based on different image transformation properties. This objective focuses principally upon refining the inpainting framework to achieve better disocclusion hole-filling, while also affording a trade-off with the overall inpainting time.

The new inpainting framework is illustrated in Figure 1.3. It delivers a robust, flexible and efficient solution to disocclusion hole-filling by consistently providing an enhanced visually plausible synthesised view. The high quality inpainting is accomplished by successfully fulfilling both objectives 2 and 3 above, by introducing and extending the concept of self-similarity characterisation. This builds upon the core block, namely *joint texture and depth inpainting*, developed as the main outcome of objective 1.

1.5 Contributions

Guided by the aforementioned research objectives, three original contributions have been made by the new inpainting framework to tackle the challenges in disocclusion inpainting process and provide an efficient solution to deliver perceptually pleasing view quality.

The three original scientific contributions made in this thesis to the multi-view inpainting domain are as follows:

1. ***Joint Texture-Depth Inpainting (JTDI)***: This introduces a novel depth-based inpainting approach for filling disocclusion holes, which simultaneously exploits both the texture image and corresponding depth map. Using the available depth information, a BG first hole-filling strategy is formulated to fill missing texture pixels and then applies this information to inpaint the corresponding depth pixels. This joint texture and depth approach results in more effective inpainting both quantitatively and qualitatively.

2. ***SC-JTDI***: This algorithm introduces image *self-similarity characterisation* (SC) at the encoder and transmits it as supplementary information to construct enhanced search space at the decoder. The potential search space is confined to the BG region and the received characterisation information helps in generating a superior search space for TM. This results in reduced visual artefacts and improved quantitative performance. The restricted BG oriented search space also means faster inpainting time in comparison to JTDI.

3. ***ASC-JTDI***: An *advanced* self-similarity characterisation (ASC) is developed as an extension to SC-JTDI which incorporates additional image characteristics to expand the search space. It employs a mechanism for flexibility that automatically determines image-specific, self-similarity characteristics and applies this information to broaden the search space for improved TM. This not only benefits the inpainting by providing more reliable characterisation information to generate an effective search space for refined inpainting, but also reduces the corresponding visual artefacts.

1.6 Thesis Structure

The remainder of the thesis is organised as follows:

- *Chapter 2* presents a brief introduction of multi-view technology followed by a comprehensive review of view synthesis techniques and a literature survey on various inpainting techniques. The critique in Chapter 2 helps to identify the gaps in existing hole-filling methods, specifically for inpainting disocclusion holes.

- *Chapter 3* explains the integrated research methodology adopted, including the aspects of idea prototyping, testing and validation. The choice of datasets, performance metrics, simulation platform and software code validation used during various stages of the thesis are discussed.

- *Chapter 4* presents the first contribution which is a depth-based solution to encourage BG to FG filling and joint inpainting of the synthesised texture

and depth views. A rigorous quantitative and qualitative performance analysis is carried out and its impact on the inpainting performance evaluated. Work from this chapter has been published in (Reel et al., 2013).

- *Chapter 5* introduces the novel concept of self-similarity characterisation for inpainting the holes. This new technique enables fast inpainting that provides superior visual and numerical performance. Work from this chapter has been published in (Reel et al., 2014).

- *Chapter 6* presents an advanced characterisation technique which incorporates additional image self-similarity characteristics for inpainting disocclusion holes. It broadens and strengthens the basic idea of self-similarity characterisation by providing reliable and focussed self-similarity information which is applied effectively to refine the inpainting process.

- *Chapter 7* discusses some potential research directions which can exploit the novel inpainting framework presented.

- *Chapter 8* makes some conclusions on the main findings and original contributions made.

1.7 Summary

This chapter has introduced the inpainting problem and motivation behind the overarching research question addressed by this thesis. Three principal research objectives have been proposed to address the major challenge of hole-filling in virtual view synthesis, with a particular focus on inpainting disocclusion holes,

which can lead to severe visual artefacts and a degraded interactive experience. The proposed inpainting framework aims to provide an effective and efficient solution for perceptually pleasing view synthesis. The next chapter presents a critical literature review of existing view inpainting techniques.

Chapter 2

Inpainting: A Review

2.1 Introduction

The chapter provides a broad overview of well-known inpainting techniques and their applicability in inpainting disocclusion holes generated during view synthesis. To provide a better understanding of various artefacts and challenges, an overview of virtual view synthesis in different scenarios is also presented. Section 2.2 and 2.3 briefly discusses the historical background of 3D and multi-view technology.

2.2 Background of 3D Technology

In 1844, David Brewster invented a stereoscope capable of taking 3D photographic images. This stimulated the research in 3D technology and in 1851 a stereoscopic picture of Queen Victoria (see Figure 2.1) became famous worldwide (Crary, 1992).

(a) (b)

Figure 2.1: shows (a) Stereograph of Queen Victoria and (b) stereoscope displaying slide of Queen Victoria (King, n.d.) .

The stereoscopic cameras caught attention and in 1890 a renowned British Film-maker 'William Friese-Greene' filed a patent of 3D movie production process (Zone, 2007). The idea was to project both left and right images (slightly different views of the same scene) on one screen and when seen through red and green glasses it resulted in 3D effect. This led to research in 3D film technology and first publically released 3D movie was 'The Power of Love' in 1922 (Kondoz and Dagiuklas, 2013).

On the other hand, the Television broadcast started gaining popularity during 1950's and thus began the experimental demonstrations of 3DTV. Several limiting factors involving transmission, storage and displays were researched and during the 1990s the MPEG started working on compression technology for stereoscopic video sequences (Schreer et al., 2005). The improvements in 3D technologies led to growing research in 3DTV and FTV (Morvan et al., 2008; Shade et al., 1998; Zitnick et al., 2004). Multi-view technology has gained increased attention from researchers in both academia and industry, aiming to enhance the immersive experience of the user through free-viewpoint viewing (Fujii and Tanimoto, 2002; Kubota et al., 2007).

(a) (b)

Figure 2.2: (a) Texture image and (b) Depth map.

2.3 Multi-view Technology

A depth-based 3D representation emerged as an efficient and flexible approach for enhanced multi-view technology (Kubota et al., 2007; Tanimoto et al., 2011; Zhu et al., 2012). The 3D representation refers to '2D + Z' format which consists of a 2D texture/colour image and its corresponding depth map as shown in Figure 2.2 (a) and (b) respectively. The depth map is basically a grey-scale image, resulted by assigning a depth value (z-value) to each pixel in the colour image.

The depth map can be generated using physical ranging methods such as time-of-flight (TOF) sensor (Stemmer Imaging Ltd., nd) and structured light scanner. An example of TOF and structured light sensor methods are shown in Figure 2.3 (a) and (b) respectively. The TOF method is based upon measuring the depth of scene-points by illuminating the scene with a controlled laser source and analyse the reflected light (Zhu et al., 2012). Ku (Leuven, nd) explains structured light as the process of projecting a known pattern (often grids or horizontal bars) on to a scene and calculating the depth and surface information of the objects by analysing the deformed pattern.

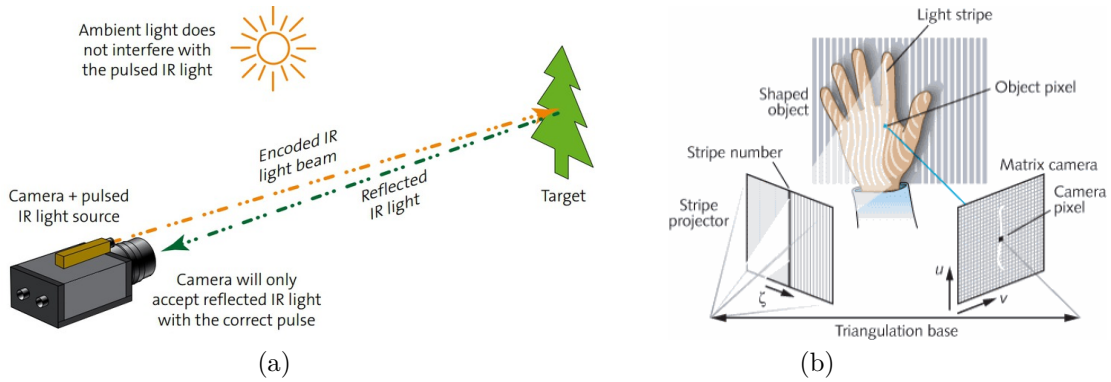


Figure 2.3: Example representing (a) Time-of-flight (Stemmer Imaging Ltd., nd) and (b) Structured light scanner (Leuven, nd).

These methods tends to have limited capture range which may results in incorrect measurements. The interference from multiple emitters is another limiting factor. In recent years, researchers have been developing these technologies to overcome the limitations for improved depth sensing (Horaud et al., 2016).

Multiple views of same scene captured in texture-plus-depth format are called Multi-view Video-plus-Depth format (Muller et al., 2008a; Smolic and Kauff, 2005; Smolic et al., 2011; Zitnick et al., 2004). These representations enable applications such as free viewpoint viewing which allows the viewer to freely navigate among the available 3D views by changing the viewpoints. One or more virtual views of the 3D scene can be synthesised in real-time at the receiver side by a technique called DIBR (Mark, 1999; McMillan, 1997; Muller et al., 2011; Tian et al., 2009; Vetro et al., 2008). The virtual views can be rendered with arbitrary baseline and the number of rendered views can be larger than the original views. To synthesise an intermediate virtual view, only a subset of required display views has to be transmitted (Kubota et al., 2007; Merkle et al., 2009, 2007; Smolic et al., 2008).

The next section discusses the virtual view synthesis in detail.

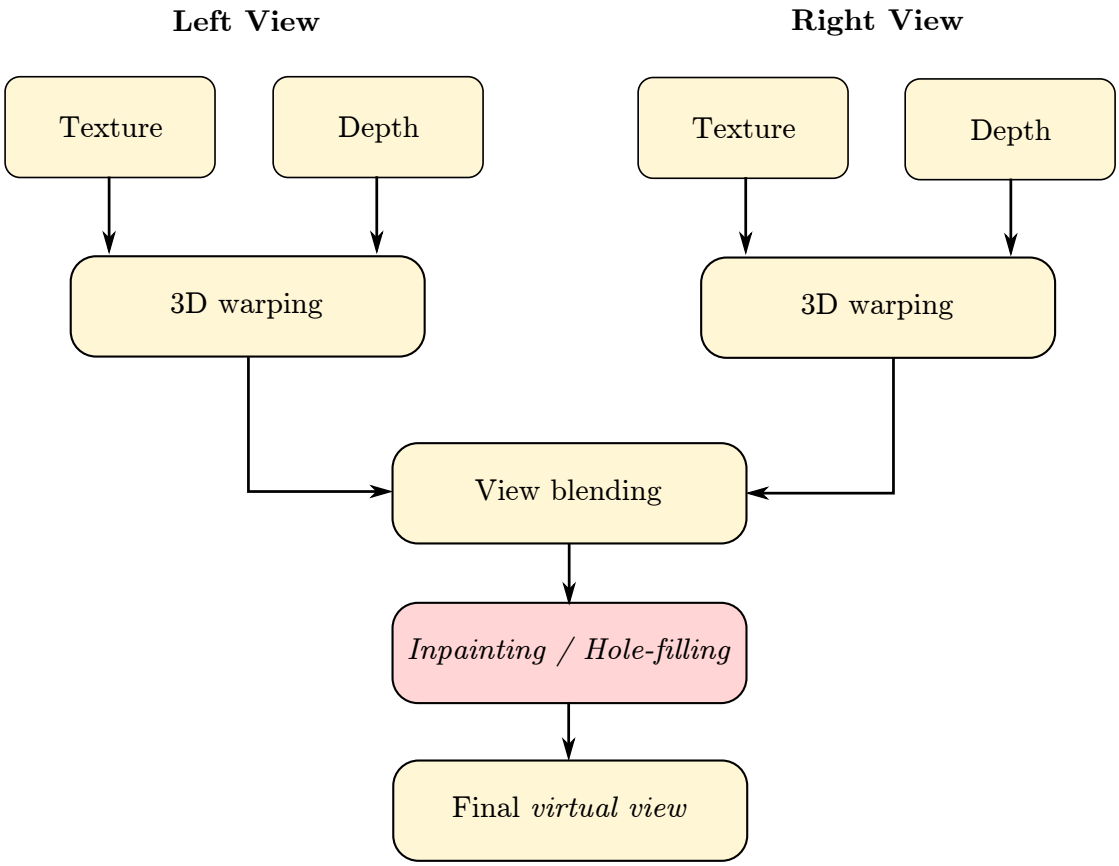


Figure 2.4: Virtual view synthesis with two reference views (DS-DIBR)

2.4 Virtual View Synthesis

Virtual view synthesis can be achieved by using one or more reference views. The view synthesis using single reference view and two reference views are termed as *Single Sided-DIBR* (SS-DIBR) and *Double Sided-DIBR* (DS-DIBR) respectively. This comprises of 3D warping, followed by inpainting / hole-filling to achieve high quality virtual view.

However, in case of DS-DIBR, an additional step of view blending (Section 2.4.2) is performed after 3D warping step, as shown in Figure 2.4.

2.4.1 3D Warping

To synthesise a virtual view, a reference texture image and its associated depth/disparity map is required (Jiufei et al., 2010; Kauff et al., 2007; Tian et al., 2009). 3D warping is a pixel-to-pixel mapping such that the reference image pixels are first projected back to the world coordinates using depth map and then reprojected to the arbitrary virtual image coordinate (Mark et al., 1997; Tian et al., 2009). To understand a warping process, consider a case where the cameras are set-up in a 1D parallel arrangement such that the two cameras are aligned and have only translational or horizontal shift (i.e. *u-axis*) but no rotational shift (*v-axis*).

For simplification, assume the reference and virtual camera share the same focal length F and rotation matrix. Such that a pixel u_r, v_r in the reference view can be mapped to u_v in the virtual view as:

$$u_v = u_r + \frac{F \times (t_{x,v} - t_{x,r})}{Z} + (o_{x,v} - o_{x,r}) \tag{2.1}$$

$$u_v = u_r + d \tag{2.2}$$

where

$$d = \frac{F \times (t_{x,v} - t_{x,r})}{Z} + (o_{x,v} - o_{x,r}) \tag{2.3}$$

is the disparity, $t_{x,v}$ and $t_{x,r}$ are the translational vector for virtual and reference views respectively, and their difference describes the baseline spacing. $o_{x,v} - o_{x,r}$ is the difference of the principle point offset for virtual view and reference views. Thus provided disparity or depth map, each pixel in a reference view can be mapped to corresponding point in the virtual view. However, pixel mapping is not one-to-one between reference and virtual view, such that multiple pixels attempt to acquire same pixel location, but the pixel closest to the camera i.e. FG pixel,

is selected and mapped to that position. In other case, a pixel may be mapped to a non-integer pixel position which is actually non-existent in the virtual view grid and so the location of projected pixel is commonly rounded to the nearest integer pixel position (Daribo and Saito, 2011; Muller et al., 2008b; Tian et al., 2009). This can lead to one pixel wide gaps called *cracks* (see Figure 1.2(e)) (Hornung and Kobbelt, 2009; Muller et al., 2011; Zitnick et al., 2004).

In practice, pixels of an object closer to the camera have larger displacements during a viewpoint change than pixels of the BG. This means that there may exist one or more spatial regions of the BG, occluded by a FG object in the reference view that becomes exposed in the virtual view (Daribo et al., 2012; Kim et al., 2011; Muddala et al., 2016). Due to viewpoint change, some pixels in the reference view are unavailable and thus never get mapped to the virtual view. These unmapped pixel positions with no corresponding pixels in the reference view are commonly called *disocclusion holes* (see Figure 1.2(d)) (Ahn and Kim, 2012; Buyssens et al., 2015; Chen et al., 2010b; Do et al., 2009; Fehn, 2004a; Gui et al., 2013).

This is elaborated using an example in Figure 2.5; a horizontal camera arrangement shows 3 cameras at positions C_1, C_2 and C_3. Assume C_2 alone is used to synthesise a virtual view at position $C_{virtual}$, the region occluded by the FG in C_2 becomes visible i.e. disoccluded in virtual view. The disoccluded region degrades the visual performance and thus needs to be filled with visually plausible information.

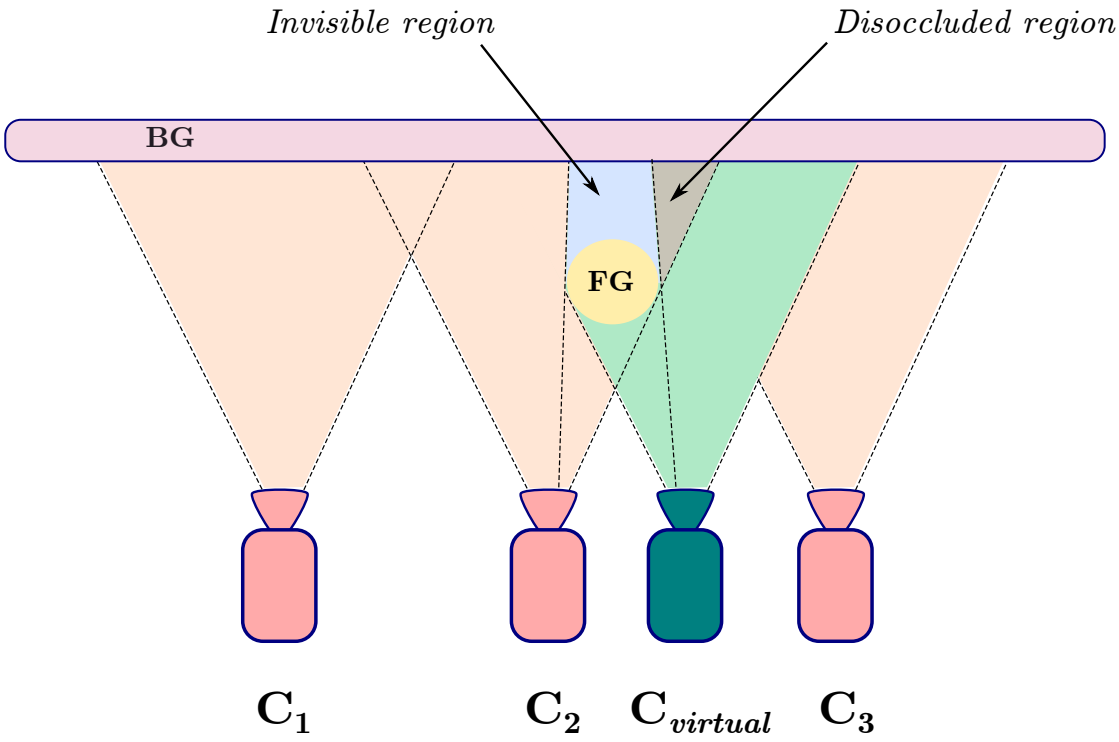


Figure 2.5: Horizontal multi-view camera set-up depicting virtual viewpoint $C_{virtual}$ and disoccluded region.

2.4.2 View Blending

In case of DS-DIBR (see Figure 2.4) scenario that is used for view generation at a given viewpoint, both left and right reference views are warped separately to synthesise two virtual views. The two warped views correspond to a same virtual viewpoint and contain certain information missing in each-other. The holes in one view can be filled with the information content from other and is known as view-merging/blending. These two warped views can be blended together by using a linear weighting function to blend pixels from two warped views (Muller et al., 2008b; Zinger et al., 2010) or consider one view as the main view and then fill the holes in selected view from the other warped view (Domaski et al., 2009; Gao et al., 2013). The amount of holes in the virtual view generated using DS-DIBR (Gui et al., 2013) is less as compared to the holes in SS-DIBR due to view blending.

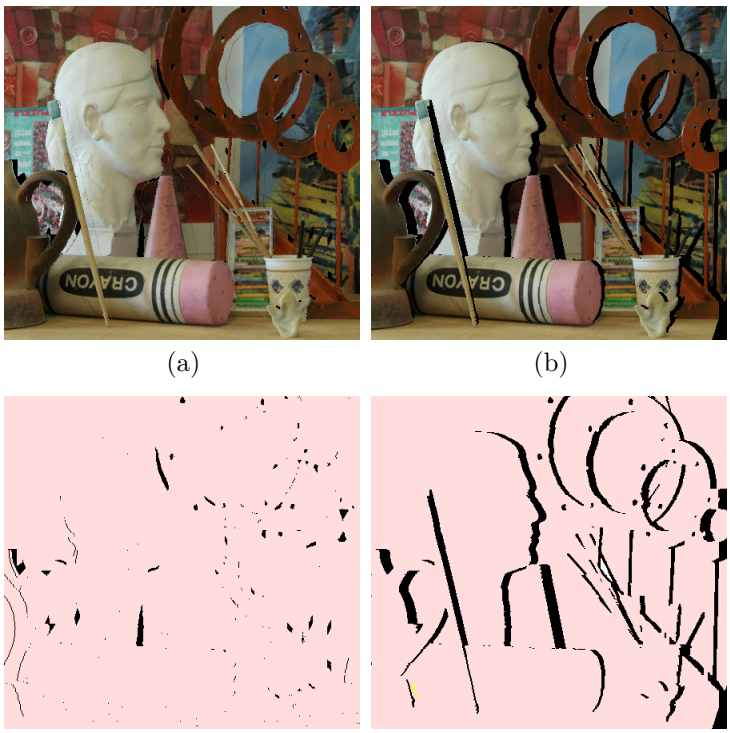


Figure 2.6: Synthesised virtual views after (a) DS-DIBR and (b) SS-DIBR; (c) and (d) represent their corresponding holes regions, respectively.

Figure 2.6 shows the comparison of holes generated using SS-DIBR and DS-DIBR for *Art* dataset (Scharstein and Pal, 2007; Scharstein and Szeliski, 2003). It is clearly evident that disocclusion problem is particularly challenging when using SS-DIBR (Lei et al., 2016).

2.4.3 Inpainting/Hole-filling

After 3D warping, two types of holes appear, namely 1) *cracks* and 2) *disocclusions*. As discussed, cracks are basically one pixel wide holes which occur due to round-off errors and can be filled using techniques like interpolation, filtering procedures (Mori et al., 2008; Muller et al., 2008a; Oh et al., 2009) and morphological operations (Ahn and Kim, 2013). The second type of holes called disocclusion

holes occurs when spatial region occluded by a closer object in the reference view, but become visible in the virtual view (Ahn and Kim, 2012; Muddala et al., 2013; Reel et al., 2013; Schmeing and Jiang, 2015; Tauber et al., 2007). Disocclusion holes typically occur at FG object boundaries (specially using SS-DIBR) and are considered a major problem when synthesising novel viewpoint images via DIBR (Daribo et al., 2012; Lei et al., 2016; Muddala et al., 2016; Xu et al., 2013; Zhu and Li, 2016).

The amount of disocclusion holes is baseline distance dependent such that as the distance between the virtual and original views increases, so does the disocclusion holes. The appearance of these holes cause visible artefacts and is perceptually unpleasing (Ahn and Kim, 2012; Do et al., 2009; Fehn, 2004b; Gao et al., 2013; Gautier et al., 2011). The technique used to fill the missing information in the image is known as *Inpainting* or *hole-filling* (Ashikhmin, 2001; Bertalmio, 2001; Bugeau et al., 2010; Buyssens et al., 2015). Inpainting large disocclusion holes to render high quality virtual views have been a major challenge in the research community. This work focuses on inpainting the disocclusion holes in the DIBR synthesised images and the next Section 2.5 discusses in detail the various techniques to overcome this problem, their advantages, limitations and scope.

2.5 Review of Inpainting Methods

Inpainting is an art of reconstructing missing or deteriorated regions of an image in an undetectable manner. The applications of inpainting range from restoration of damaged paintings and photographs to the removal or replacement of selected

objects (Bertalmio et al., 2000). The missing regions in an image are basically sets of pixels which may or may not be contiguous (Varzi and Vieu, 2004). These sets of pixels are called artefacts, scratches, gaps, and holes or unknown regions depending on the area of application. The techniques used for digital inpainting include analysis and usage of pixel information from the surrounding area to fill-in part of image (Bertalmio et al., 2001).

There are various approaches to perform inpainting but mostly the algorithms are based upon one or more of the basic techniques such as geometry-based, sparsity-based and patch-based/texture-based methods (Arias et al., 2009). The geometry-based methods are mainly based upon PDE and focuses on exploiting the geometric structure of an image to fill the missing information (Ballester et al., 2001a; Bertalmio et al., 2000; Chan et al., 2002), whereas the patch-based methods employ image content from the neighbourhood region of the missing regions in the image (Cao et al., 2011; Martanez-Noriega et al., 2012). The sparsity-based methods use sparse image representation to synthesise the missing part and optimise it using sparse distribution. Some techniques that combine both geometry and texture-based methods are known as exemplar-based inpainting methods. Following sub-sections provide a review on various inpainting techniques:

2.5.1 Geometry-Based Methods

These methods formulate inpainting as a heat diffusion problem by introducing smoothness priors to propagate local structures from the exterior to the interior of the hole. The method is robust such that it simultaneously fills the holes by con-

sidering the information around the holes but slows down the process as well cause the blur within the occluded area when the hole area is large. Thus, the algorithm is not suitable for filling large holes or reconstruction of sharp edges. A fast inpainting method based on stopping the diffusion process at certain pre-defined positions were used (Oliveira et al., 2001) which however appeared as additional user intervention. Another method combined diffusion with anisotropic filtering to have an interpretation as fluid transportation using Navier-stokes equations for fluid dynamics (Bertalmio et al., 2001) to speed up convergence.

Some inpainting methods involved a complicated energy functional which assumes bounded variation (Masnou and Morel, 1998) and total variation models (Chan and Shen, 2001) for properly reconstructing curved regions. Inpainting problem was then redefined to consider curvature in the form of Euler elastic curve to reconstruct contours of missing objects(Ballester et al., 2001b). An inpainting method called Curvature Driven Diffusion was introduced, where the amount of diffusion applied is based on the amount of isophote curvature at that point (Chan et al., 2002). This prevented the blurriness in the smooth areas and shows good improvement from Bertalmio's algorithm. Another technique used variational methods along with the higher order PDEs (Ballester et al., 2001b) to jointly interpolate the image gray-levels and gradient/isophotes directions and smoothly extend the isophote lines into the holes of missing region. Some methods allow information propagation from outside to inside the holes via a structure-preserving diffusion method (Tschumperle and Deriche, 2005). These methods provide good results when filling small regions e.g. straight lines, curves etc. However they tend to introduce blur when the missing regions are large. Also these methods are helpful

only in reconstructing the structure of the missing region but fails to recreate their texture (Kwatra et al., 2003; Ndjiki-Nya et al., 2008).

2.5.2 Sparsity-Based Methods

A sparse representation method fills the missing region using sparse combination of a redundant dictionary constructed by source patches (Elad et al., 2005; Mairal et al., 2008a,b; Shen et al., 2009). An extension to this method is made by introducing more regulation terms (Xu and Sun, 2010). Another method casts the inpainting problem into low-rank matrix recovery and completion problem (Wang and Zhang, 2011). Based on low-rank assumption, missing region recovery is formulated as a convex optimisation problem via block nuclear norm. This method promotes blockwise low-rankness of an image with missing regions (Ono et al., 2012). To recover a low rank matrix, another method formulates the problem as a Schatten-p norm minimisation problem based on the FOCally Under-determined System Solver approach (FOCUSS) (Majumdar and Ward, 2011; Majumdar et al., 2012).

In another method a texture image is modelled as 2D Autoregressive model and inpainting problem is formulated as minimising the rank of a Hankel matrix (Ding et al., 2007; Sznaier and Camps, 2005). To overcome the complexity and find an approximate solution, the nuclear norm heuristic based algorithms are proposed (Mohan and Fazel, 2010, 2012). The nuclear norm minimisation is formulated as semi-definite programming and can be solved by interior point method but it is computationally costly for a large size problem. A fast algorithm based on l_2 norm

minimisation was proposed to find a sparse vector included in the null space of a matrix (Takahashi et al., 2011) and this algorithm is based on the reweighted least squares for sparse recovery method (Daubechies et al., 2010). The performances of sparsity based inpainting approaches are highly dependent on the choice of dictionary and provide effective results only if the missing region is small such as sparsely distributed noise over the image. However, sparse based methods provide inefficient results when the missing region is large (Buyssens et al., 2015; Efros and Leung, 1999).

2.5.3 Texture-Based Methods

Broadly texture-based methods can be classified into two categories: pixel-based and patch-based synthesis. Many approaches focus on recovering texture of the missing region based on the source image. Texture synthesis approach for filling the holes with the known information can be regarded either as parametric or non-parametric models. The parametric synthesis fills the missing information using a compact model with a fixed (Heeger and Bergen, 1995; Portilla and Simoncelli, 2000) or dynamic parameter set. Non-parametric methods usually formulate the problem based on Markov Random Field (MRF) and can further be classified as sample based (De Bonet, 1997) and patch based methods (Ashikhmin, 2001; Kwatra et al., 2003; Ndjiki-Nya et al., 2008). Nonparametric methods yield better results in comparison to parametric algorithms, also they can be employed to large variety of textures (Kwatra et al., 2003). Initially a nonparametric method for texture synthesis based on MRF was proposed where a new image grows outward from an initial seed, one pixel at a time. Pixel based synthesis yielded good

results but at expense of computational cost. But for better preservation of local structures, faster and real time algorithms are required. To speed up the process instead of copying a pixel, entire patch is copied (Efros and Freeman, 2001).

Patch based texture synthesis is based on TM which copies a fixed-size repeating pixel patch from a known spatial region to the hole region. The main idea behind the patch synthesis is based on self-similarity priors (Buades et al., 2005). The computational cost was reduced by reducing the search space such that to find the best candidate patch instead of using whole image, only the neighbourhood region around the missing pixels is considered (Ashikhmin, 2001). Some other methods reduce the search space and computational cost by reducing the dimensionality of the patches with techniques like Principal Component Analysis or randomised approaches (Lefebvre and Hoppe, 2006), and employing a multi-scale framework and organising the image patches in tree structures (Wei and Levoy, 2000). These approaches maintained the coherency of synthesis and yielded good results, however it failed to preserve local structures or geometry within an image.

Since real images have both texture and structure content, neither geometry nor texture-based methods alone can offer an adequate solution. For the preservation of local structures as well as the composite textures, the advantages of both structural and textural inpainting are combined, making it possible to reconstruct both texture and geometric structures (Aujol et al., 2010; Kawai et al., 2009; Wexler et al., 2004, 2007). Such techniques which combine structure and texture synthesis is known as exemplar-based inpainting techniques and (Criminisi et al., 2004) is regarded as very significant work in the field of the image inpainting.

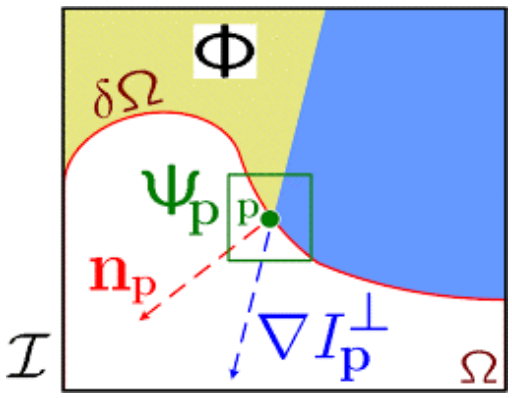


Figure 2.7: Diagram representing notation used for Exemplar-Based Inpainting (Criminisi et al., 2004).

2.5.4 Exemplar-Based Inpainting

An exemplar-based inpainting technique aims to employ a patch priority scheme to determine which patch to fill first followed by its texture inpainting. The missing pixels at the edge of an image object have higher priority than missing pixels on flat regions. This priority is calculation is based on *confidence* term and *data* term (Criminisi et al., 2004). The confidence term gives high priority to those patches at the edge which have more known (filled) pixels around it. The data term is a function of the strength of isophotes hitting the front $\delta\Omega$ at each iteration. This algorithm handles large fill areas which combines the use of texture synthesis and isophote driven inpainting by a priority based mechanism. This technique is most influential work in the field of exemplar based image inpainting and the details of this method and various terms are provided below:

The source region (i.e. the known region) is defined as $\Phi = I - \Omega$, where I and Ω are input image and hole region, respectively. The boundary of hole region is defined as $\delta\Omega$ and the hole region Ω may not be a single contiguous spatial region (see Figure 2.7 for an illustration).

The square pixel patches Ψ_p centred at pixel p at the border the hole region are inpainted in the order of their priority (to be discussed). Consider a specific pixel patch $\Psi_{\hat{p}}$ of default size $w \times w$ from among all the Ψ_p on the boundary, such that it has maximum priority. This patch with highest priority is termed as the target patch (TP). The best matching patch $\Psi_{\hat{q}}$, known as candidate patch (CP) is identified in the source region that is most similar to $\Psi_{\hat{p}}$ and minimises the matching error as:

$$\Psi_{\hat{q}} = argmin_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q) \tag{2.4}$$

where $d(\Psi_{\hat{p}}, \Psi_q)$ is the Sum of Squared Differences (SSD) between corresponding known colour pixels of the two patches. In other words, known pixels in $\Psi_{\hat{p}}$ are used as a template to find a best matched patch in source region. After $\Psi_{\hat{q}}$ is identified using (2.4), missing pixels in target patch $\Psi_{\hat{p}}$, $\Psi_{\hat{p}} \cap \Omega$, are filled using corresponding pixels in $\Psi_{\hat{q}}$. The order in which the missing pixel patches in the Ω are filled, is considered critical. Thus a priority term is derived that calculates the confidence and the data term, the patch with highest priority should be inpainted first.

The priority term *P(p)* for each boundary patch, where $p \in \delta\Omega$, is computed as the product of two terms:

$$P(p) = C(p) \times D(p) \tag{2.5}$$

Where $C(p)$ and $D(p)$ are the confidence and data terms, respectively. $C(p)$ and $D(p)$ are defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \Phi} C(q)}{|\Psi_p|} \tag{2.6}$$

$$D(p) = \frac{\left|\nabla I_p^{\perp} . n_p\right|}{\gamma} \tag{2.7}$$

Where $|\Psi_p|$is the number of pixels in target patch Ψ_p, γ is a normalisation factor (e.g., $\gamma = 255$ for a typical grey-level image), n_p is the unit vector orthogonal to $\delta\Omega$ at pixel p , and $\nabla I_p^{\perp}$ is the isophote (direction and intensity) at pixel p. The confidence term $C(p)$ gives higher priority to the patches which have higher percentage of non-hole pixels. $C(p)$ is initialised to 0 for missing pixels in Ω, to 1 everywhere else. $D(p)$ defines the strength of linear structures hitting the boundary $\delta\Omega$ at each iteration, and is used to encourage propagation of linear structures. After missing pixels in a patch $\Psi_{\hat{p}}$ are filled, the confidence term $C(p)$ for each newly filled pixel p in the patch is updated as follows:

$$C(p) = C(\hat{p}), \forall p \in \Psi_{\hat{p}} \cap \Omega \tag{2.8}$$

The confidence values are updated, priorities for the next patch to be filled are computed and this entire process is repeated till all disocclusion holes are filled.

Improvement to this exemplar-based method was proposed to change the fill order and a matching cost function (Nie et al., 2006). Unlike the original method where data term and confidence terms are multiplied for priority calculation (i.e. if data term is zero then priority also becomes zero), the data and confidence terms are added. Another method involved a nonlocal-means approach to infer the target patch by weighting a set of similar candidate patches (Wong & Orchard, 2008). Some methods copy multiple patches in a single step and thus are termed as

greedy approach (Bornard et al., 2002; Drori et al., 2003; Martanez-Noriega et al., 2012; Meur et al., 2011) to speed up the process. Several attempts were made to improve the priority by using a tensor based data term (Meur et al., 2011); magnifying data term (Martanez-Rach et al., 2014); limiting search space through user-intervention (Sun et al., 2005).

Since exemplar methods are based on self-similarity prior, some methods broadened the idea by considering various transformations such as varying the scales of patches (Drori et al., 2003); extending the search space by varying possible scale and rotations of source patch (Barnes et al., 2010; Mansfield et al., 2011). But this required huge computation complexity thus another method was proposed restricting the search space by minimising Euler's elastica of contrasted level lines (Cao et al., 2011). Another method proposed detection for transformed patches (Fedorov et al., 2016, 2015; Huang et al., 2014). The exemplar based methods proved to establish better results in comparison to the previous methods. In multi-view imaging system depth is an additional feature and recently it has been employed to aid in inpainting the missing texture region (Ahn and Kim, 2012, 2013; Daribo and Pesquet-Popescu, 2010). The next section showcases various depth based image inpainting techniques.

2.5.5 Depth-aided Inpainting

The inpainting techniques discussed above are insufficient for filling disocclusion holes and requires improved strategy to fill the missing regions. The priori knowledge about disocclusions is that they are result of displacement of FG object

revealing BG areas (Tauber et al., 2007). Therefore, disocclusion holes are located on the border between FG and BG, and are required to be filled with the neighbourhood located on the BG rather than FG. Applying the non-depth assisted methods, as discussed above, tends to wrongly fill the missing regions with both the BG and FG pixel information and thus cause considerable visual artefacts (Chen et al., 2010a; Gui et al., 2013; Jantet et al., 2011; Oh et al., 2009). To minimise the artefacts some methods process the depth maps to eliminate the holes before employing it to fill the holes in the texture image (Cheng et al., 2011; Koppel et al., 2010; Ndjiki-Nya et al., 2011) and other methods employ depth to distinguish FG from BG to inpaint the missing region from BG only region (Oh et al., 2009). Certain methods tends to use both the techniques to perform the inpainting. The various methods used in the literature to perform depth-aided inpainting are discussed now.

As most of the missing information belongs to the BG, segmentation (Silva et al., 2010) is performed which involves classifying the data into FG and BG, and then adequately inserting the BG samples into the disoccluded region. Various techniques like interpolation or simple image inpainting methods are used to perform inpainting but they tend to introduce blur in the unknown areas (Ndjiki-Nya et al., 2011). One method is to repeat line-wise the last valid BG sample into the missing region (Muller et al., 2008b). This technique performs poorly when applied to inpaint structured BG and dominant vertical edges. Another method fills the missing region with texture synthesis (Jiufei et al., 2010) but that led to blocking artefacts or luminous inconsistencies in the virtual view. Some methods perform pre-processing of depth maps to smooth the depth data across the

edges and this lowers the depth gradients in the virtual view. Most often used filters are Gaussian low-pass filter (Fehn, 2004b) or asymmetric filter (Zhang and Tam, 2005). Using this approach usually distorts the FG objects which affect the output view (Chen et al., 2005). Some methods combine both the approaches i.e. pre-processing the depth map with a bilateral filter and then filling the texture using the available BG information (Cheng et al., 2008). Another similar method uses edge-dependent Gaussian filter and fills the remaining holes via edge-dependent interpolation (Chen et al., 2010a). These approaches partially eliminate the geometric distortions but leads to increased computational complexity (Solh and AlRegib, 2012).

Some methods exploited temporal consistency across successive frames to improve the inpainting (Koppel et al., 2010; Yao et al., 2014). Another approach extracted the BG information first by background subtraction and then filled missing region (Schmeing and Jiang, 2015). However, this caused omission of illumination variation compensation and also required manual correction of disocclusion. Another approach for handling disocclusions considers statistical dependencies between different images of a sequence via a BG segment. The holes are first coarsely estimated and then refined using texture synthesis (Ndjiki-Nya et al., 2011). The drawback in this approach is the depth estimation inconsistencies may lead to considerable degradation.

Some methods synthesised stereo images from pair of stereo images using depth, edge and image segmentation to provide more information for improved filling and reduced visual artefacts. The best candidate for disoccluded regions is selected with Conditional Random Fields and graph-cuts (Scharstein and Pal, 2007; Tran

et al., 2010). Other method exploited disparity information and inter-view correlation instead of using graph cuts optimisation and mean shift segmentation (Jain et al., 2011). A similar method uses disparity values to separate hole positions present in FG and BG layers and fill in the missing information (Ramachandran and Rupp, 2012). Techniques like Hierarchical Hole-Filling minimises the geometric distortion by using pyramid structure for lower resolution estimation of 3D warped image to help in estimate the hole pixels (Solh and AlRegib, 2012). Some approaches using layered depth images or multiple reference texture and depth images to fill in the holes tends to be computationally expensive (Wang et al., 2015).

One approach is to replace the FG boundaries with the BG ones located on the opposite side by intentionally manipulating disocclusion boundaries to have pixels only from BG (Oh et al., 2009) and then to apply existing inpainting techniques. But this method does not fully inpaint the holes with the BG since FG boundaries are not always well identified and properly replaced. Another method extended Criminisi's algorithm by modifying the priority function and giving higher priority to BG pixels over FG (Daribo and Pesquet-Popescu, 2010). The higher priority is given to the patch with lower depth variance.

Assuming depth information is available per pixel in the entire virtual view, an extra term $L(p)$ was added to $P(p)$ in (2.5):

$$P(p) = C(p) \times D(p) \times L(p) \tag{2.9}$$

where $L(p)$ is a depth variance term, proportional to the inverse variance of the corresponding depth patch Z_p:

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum_{q \in Z_p \cap \Phi} (Z_p(q) - \overline{Z}_p)^2} \tag{2.10}$$

where $|Z_p|$ is the size of depth patch Z_p, $Z_p(q)$ is the pixel depth value at the pixel location q under $\overline{Z}_p$ which is mean depth value. The results with this method contain noticeable errors as the assumption that patches of low depth variance belong to BG is not always true. Also sometimes the boundaries of objects in the depth map are mismatched with that of a colour image. The other extension of Criminisi's algorithm defined data term using 3D structure tensor of Di Zenzo matrix and also add depth information in the best patch calculation module (Gautier et al., 2011). But the problem is Di Zenzo matrix reflects only the strong gradients well. Also this method tends to introduce blur as it combines five best patches to fill target regions.

In order to improve upon these shortcomings another method based on the Criminisi's technique used Hessian matrix structure tensor and epipolar line term. The best matched patch is selected considering the data term on the BG regions which is extracted using warped depth map (Ahn and Kim, 2012, 2013). However, this tends to provide inferior results in case of intermediate FG objects (Cheung et al., 2015).

Another improvement to exemplar-based technique, apart from adding depth information to the priority function, applies local segmentation to prevent propagation of FG objects into BG texture. Then a gradient based searching is performed to lower the computational cost by adapting the search window size (Ma et al., 2012). The inpainting artefacts occur if segmentation fails to properly separate the

BG and FG. Another recent approach uses depth for estimation of scene geometry (Kohli et al., 2012). Some methods improve the accuracy of patch matching by using location distance as a penalty (Ma et al., 2012). Another method use gradient information as auxiliary information while searching the optimal matching patch (Wang et al., 2015). However to increase the computation speed a parallel computing platform is used which included Graphic processing unit to attain 51-fold faster computation (Kuo et al., 2013, 2015). These methods provide improvement over the non-depth assisted methods, but there is scope for improvement. Next section provides a discussion on existing disocclusion hole-filling methods and their limitations.

2.6 Discussion

As discussed in Section 2.5.4, exemplar based approach has been chosen for inpainting disocclusion holes due to its structure and texture preserving properties. However, due to certain characteristics of disocclusion holes (e.g. requirement to be filled from the BG information), the classic exemplar-based techniques does not work well and the visual quality of synthesised view is compromised. Having the *a priori* information about location of disocclusion holes, depth has proved helpful in framing the inpainting strategy for disocclusion holes.

The well-known exemplar-based method (Criminisi et al., 2004) is based upon a priority term, numerous subsequent works (Ahn and Kim, 2013; Daribo and Pesquet-Popescu, 2010; Gautier et al., 2011) kept the TM framework but modified the definition of the priority term and/or the criteria for TM, using available

depth information. Some methods modified the priority term to initiate the filling from the BG holes towards the FG and by distinguishing the FG and BG such that the best matching patch shall be selected from the BG only. Depth-assisted techniques helped in improving the inpainting of disocclusion holes, though the problem remains how to ensure priority selection from BG.

Some methods proposed depth pre-processing to eliminate first the depth holes by using filtering techniques which tends to introduce blur and often resulted in additional steps before the actual texture image could be filled. Few others assumed availability of pre-filled depth maps to assist filling of texture holes which is quite impractical in real-time scenarios.

The synthesised view contains holes in both the texture and depth maps, thus a more practical approach is to simultaneously inpaint the missing texture and depth information and provides consistent texture and depth inpainting. However, little attention has been paid to jointly fill both texture and depth maps. One method proposed to fill the depth holes simply by copying the depth values from the candidate depth map corresponding to the candidate texture image. This direct copying of depth pixels solely on the basis of texture seems a tenuous proposition and needs further investigation.

Table 2.1 highlighted major existing inpainting techniques that mainly focusses on disocclusion holes along with their limitations. Depth is an additional information available in DIBR process and recently it has been utilised for inpainting the disocclusion holes. However, it needs to be explored further to effectively prioritise the filling order and perform a joint inpainting of texture and depth holes.

Method	**Depth-assisted method**	**Use pre-processed depth or pre-filled depth**	**Texture inpainting**	**Simultaneous texture-depth inpainting**	*Limitations*
(Criminisi et al., 2004; Fedorov et al., 2015)	×	-	✓	-	Tends to introduce artefacts by propagating FG into BG in filling synthesised views.
(Ma et al., 2012; Muddala et al., 2013; Solh and AlRegib, 2012)	✓	×	✓	×	Improves texture inpainting but does not inpaint depth holes.
(Cheng et al., 2011; Daribo and Pesquet-Popescu, 2010; Gautier et al., 2011; Koppel et al., 2010; Lei et al., 2016; Ndjiki-Nya et al., 2011; Oh et al., 2009; Ramachandran and Rupp, 2012; Wang et al., 2015; Xu et al., 2012)	✓	✓	✓	×	1. Require pre-processed or pre-filled depth. 2. No simultaneous texture and depth filling.
(Ahn and Kim, 2012)	✓	×	✓	✓	1. Inferior results in case of intermediate FG objects. 2. Depth filling method needs further investigation.

Table 2.1: Comparative summary of major existing inpainting methods.

Overall these observations conclusively confirm the need of a new inpainting framework which effectively uses available depth information to simultaneously inpaint both texture and depth disocclusion holes. This discussion highlights the scope of improvement for inpainting technique which can provide superior hole-filling to provide enhanced view synthesis.

2.7 Summary

This chapter briefly discussed the view synthesis scenarios and highlighted the role of inpainting during view rendering. It provided a thorough literature review on various inpainting techniques, their advantages and limitations for filling disocclusion holes. The exemplar-based inpainting technique has been most popular and a critical analysis of various extensions made to this work have been discussed. The detailed review has been presented for depth-assisted methods for inpainting disocclusion holes. Based on the literature review, a summary table is provided which highlights the major existing techniques with a discussion on their limitations. The next chapter presents the adopted research methodology for this thesis.

Chapter 3

Research Methodology

3.1 Introduction

In a virtual view synthesis scenario, inpainting the disocclusion holes is highly challenging. This chapter presents the research methodology adopted to address the challenge. Various strategies involved in design, development and critical evaluation of proposed inpainting framework will be discussed in detail. A wide range of image datasets are prerequisite to test the robustness of the framework. However rigorous evaluation of the performance using justifiable quantitative and qualitative comparison metrics is equally important. From the implementation to the validation of the software code, each component is essential in the framework development.

The following sections will discuss all these parameters involved in building the inpainting framework in detail.

3.2 Research Methodology and Test-bed

To achieve an effective and efficient inpainting solution that provides a perceptually pleasing virtual view, this section explains the various phases of the research methodology adopted and details the experimental test-bed used for critical evaluation of the new algorithms developed. The key steps involved in design, development, implementation and validation of the proposed inpainting framework can be summarised as follows:

1. Perform critical literature review by exploring the recent and state-of-art inpainting techniques used for inpainting in DIBR synthesised virtual views. Identify their limitations and analyse the key assumptions relating to existing inpainting methods.

2. Implement established DIBR scenarios (SS-DIBR and DS-DIBR), which will be used as an underlying block for synthesising intermediate virtual views. Figure 3.1 shows a high-level block diagram of the adopted research methodology. It involves the pre-processed virtual view with holes and multi-view test datasets as an input to the inpainting framework. The dashed block represent various steps involved in development of inpainting framework. The output of the inpainting framework results in final inpainted view.

3. Develop a software simulation based test-bed for developing and testing proposed inpainting framework for disocclusion inpainting. Software implementation offers quick development and verification cycle in comparison to the hardware based solution. The simulation platform provides great flexibility to encompass additional functionality and is cost effective as well.

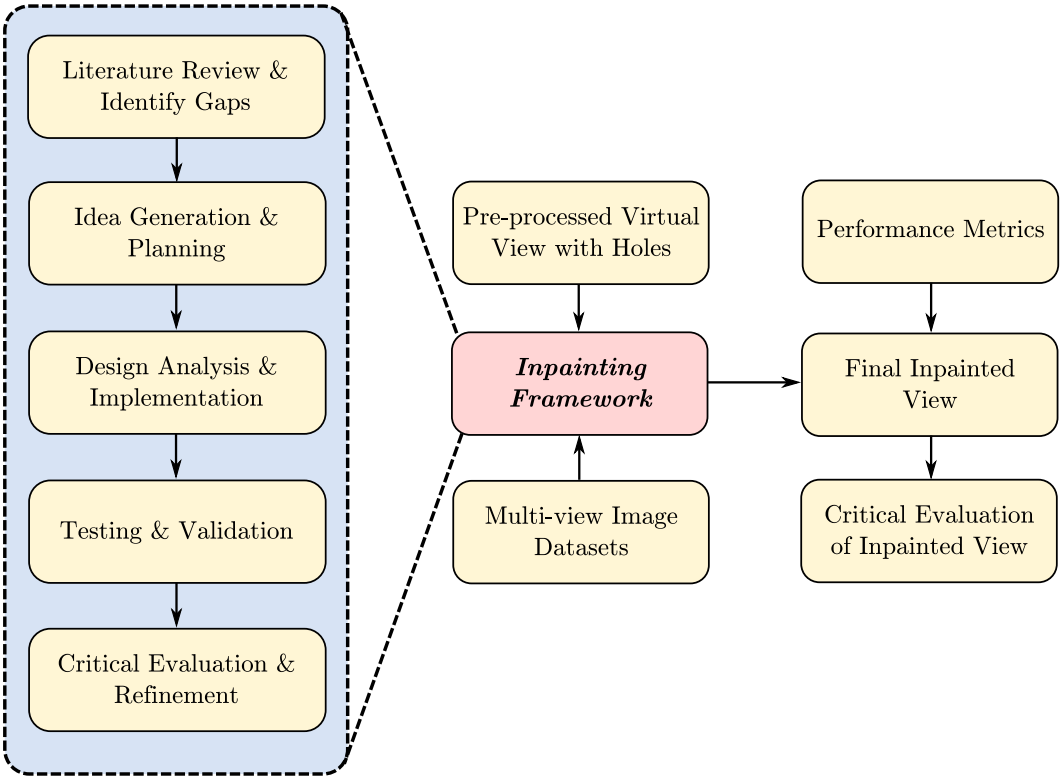


Figure 3.1: Research methodology adopted for inpainting framework.

4. Before undertaking critical evaluation of each contribution, software code is validated by performing rigorous testing.

5. The robustness of proposed inpainting framework is tested by identifying and employing different multi-view datasets commonly used by the research community. These datasets provide ground truth and are widely adopted by community to benchmark various inpainting algorithms quantitatively and qualitatively.

6. Critically evaluate the performance of proposed framework by comparing with established inpainting methods in terms of the quality of synthesised view using appropriate performance metrics.

7. Steps 3-6 are repeated and a critical analysis performed to fulfil the research objectives.

These steps provide necessary rigour to critically evaluate the inpainting framework. The next section will discuss the view synthesis scenarios.

3.3 View Synthesis Scenarios

Virtual view synthesis from an array of cameras helps provide interactive viewing. The number of cameras used in a free viewpoint television is generally a trade-off between data amount and rendering quality (Tola et al., 2009). From rendering quality perspective, disocclusion hole poses the major challenge for efficient view synthesis. The number of disocclusion holes occurring in the synthesised view is highly dependent on both the baseline distance between the reference and virtual viewpoint and on the adopted view synthesis scenario. This thesis employs two different view synthesis scenarios, namely DS-DIBR and SS-DIBR as discussed in Chapter 2.

For view synthesis, a normalised baseline is considered where the left and right cameras are set at positions 0 and 1, the virtual camera is positioned at α where $0 < \alpha < 1$ (Ramachandran and Rupp, 2012). DS-DIBR utilises view #1 and view #5 (i.e. V1 and V5) and generates three intermediate virtual views namely V2, V3 and V4 at α =0.25, 0.5 and 0.75, respectively. The SS-DIBR, uses only V1 to generate V3 at $\alpha = 0.5$. The amount of disocclusion holes in SS-DIBR is more as compared to DS-DIBR and hence more challenging to fill. In Chapter 4, the inpainting technique has been tested and evaluated on views synthesised using both DS-DIBR and SS-DIBR respectively. The experiments undertaken in Chapters 5 and 6 involve SS-DIBR scenario to test the respective inpainting

performance. The synthesised view is pre-processed to fill the cracks, i.e. single pixel holes, through interpolation before performing the disocclusion hole-filling as discussed in section 2.4.3.

3.4 Image Datasets

The testing and evaluation has been carried out on the *Middlebury 2003, 2005* and *2006* datasets (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003). Each dataset in *Middlebury 2005 & 2006* consists of 7 captured colour/texture views of same scene as well as the disparity maps for view #1 and view #5 as shown in example Figure 3.2. The Middlebury 2003 datasets, namely *Teddy* and *Cones*, consist of nine texture views including disparity maps for view #2 and view #6.

The disparity maps represent the 'inverse depth' because the disparity is inversely proportional to the depth, however features such as object edges, remain the same (Tosic et al., 2011). The disparities are expressed in rectified two-view geometry and are also called 'projective depth' (i.e. 3D scene reconstruction) (Scharstein et al., 2014) and thus the term disparity image is often referred to as depth map in the literature (Lu et al., 2012; Solh and AlRegib, 2012). Without loss of generality therefore, throughout the remainder of the thesis, the term 'depth map' is used when referring to the disparity image.

The images sequences are captured from equally-spaced viewpoints along the x-axis from left to right. This can be seen clearly in Figure 3.2 by closely observing

Figure 3.2: *Art* representing 7 camera captured texture views with depth maps for view #1 and view #5 (Scharstein and Pal, 2007).

the jug area reducing along the horizontal shifting views from view #0 to view #6. The images have been rectified to provide a pure horizontal image motion.

The wide range of datasets covers variety of image characteristic e.g. images with pattern as well as smooth regions, complex textures, multiple objects and illumination variations etc. Figure 3.3 shows the texture and depth image of *Aloe* from the Middlebury 2005 datasets (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003). A variety of test datasets has been used at various stages during the experimentation to showcase the strengths of the proposed framework. The images of these Middlebury datasets used in this thesis are included in Appendix A (shown as Figure A.1 and A.2). The choice of datasets during the experimentation is based on those which pose large disocclusion holes as well as complex textures. To fill-in the texture efficiently around the complex object edges without visually disturbing artefacts is challenging. The complex textures are difficult to fill but are usually less evident as compared to the errors occurring in filling smooth regions. For example, a slight variation in illumination of inpainted region on smooth areas may result in clearly visible artefact.

The images with patterns (as shown in Figure 3.3(a)) are difficult to inpaint

in particular since a small variation in the filling the repetitive pattern cause error propagation and hence result in wider artefact which causes visually disturbing view. Thus addressing a wide variety of images and filling their disocclusion holes will test the robustness and reliability of the inpainting framework.

The following three key factors governed the selection of the Middlebury datasets for testing and evaluation purpose in this thesis:

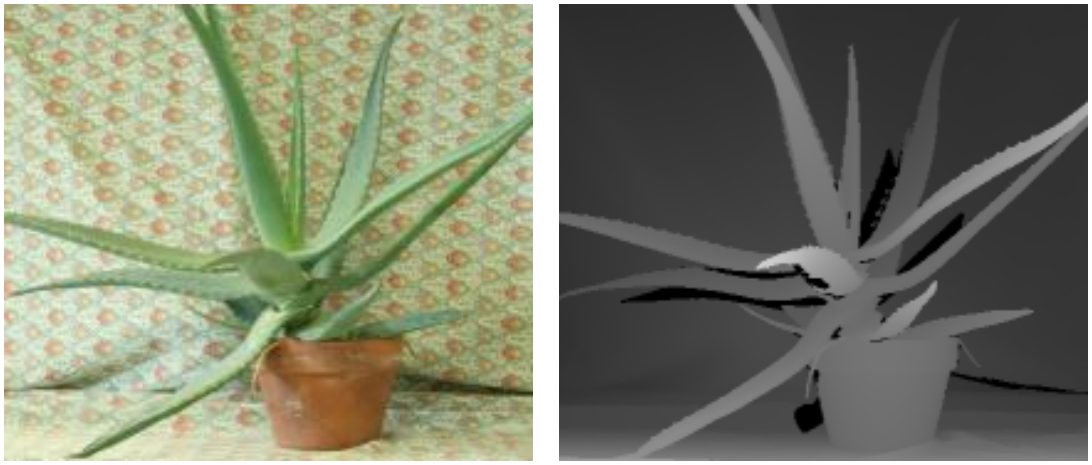
1. The availability of ground truth images is essential for both quantitative and qualitative assessment of the rendered virtual view. The Middlebury datasets provide high quality images which are used as ground truths for comparative analysis.

2. The range of these datasets provides good quality stereo sequences with highly complex geometries. Testing the proposed framework on variety of datasets with different complexities help in evaluating the robustness of the proposed inpainting framework.

3. These datasets are widely accepted by the research community for experimental analysis of the DIBR inpainting techniques (Scharstein and Pal, 2007).

3.5 Simulation Platform

In this thesis, the algorithms are designed and implemented in MATLAB® 7.14 R2012a. MATLAB is a high-level language and provides a great facilitating environment for numerical computation, visualisation, developing and prototyping



(a) Texture (b) Depth

Figure 3.3: Texture and depth image of *Aloe* from the Middlebury 2005 datasets (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003).

algorithms. Other high level languages such as *C/C++* and *Java* are compiler based and require development of components and libraries, which can be time consuming. MATLAB provides mathematically robust build-in routines, image processing toolboxes and data visualisation techniques for plotting graphs and thus makes it an obvious choice over *C++*. Open-source platforms like *Python* also support specific libraries for image processing applications but lack the necessary technical support. However, MATLAB provides an efficient and highly-reliable online support network. MATLAB exhibits much faster development time which compensates for its lower runtime performance in comparison to *C++*. MIT Lincoln Laboratory has developed *p*MATLAB that enables parallel programming with MATLAB (Kim et al., 2011) which has been used for saving inpainting time. The other technical specifications for the personal computer (PC) are detailed in Table 3.1.

Software Platform	**PC Specifications**	
MATLAB 7.14 R2012a	*Processor*	Intel® Core^{TM}(M) i5-2400 CPU @3.10GHz [Quad Core]
	RAM	4 GB, DDR2, 800MHz
	Hard Disk	250GB, 816MB Cache, 7200 RPM
	Operating System	Ubuntu 12.04 LTS (*Precise*) Kernel Linux 3.11.0-13-generic GNOME 2.30.2

Table 3.1: Simulation platform specifications and their details.

3.6 Performance Metrics

The evaluation and validation of synthesised images is a challenging task as the generated images suffer from various types of artefacts e.g., synthesis distortion (Fang et al., 2014), textural and structural distortions due to poor quality depth maps (Farid et al., 2014; Merkle et al., 2009). There is no common method formulated for evaluating inpainting algorithms in the literature (Oncu et al., 2012). The intent here is to test the performance of the inpainted texture and depth maps which is perceptually pleasing with minimal artefacts in comparison to the available ground truth. To validate the performance of proposed inpainting framework, the quality of the rendered views is measured both qualitatively and quantitatively. Furthermore an inpainting time analysis has also been performed for the developed inpainting techniques.

3.6.1 Quantitative Assessment

The quantitative assessment of inpainted virtual view is performed by means of standardised objective numerical metrics. In the literature, the most widely used image quality metrics for evaluation of inpainted images are the *Mean Squared error* (MSE) and the corresponding distortion metric, *Peak Signal-to-Noise Ratio* (PSNR) (Gonzalez and Woods, 2008; Wang and Bovik, 2009). These are proven and commonly used methods for measuring quality and analysing the similarity between the rendered and the original image. PSNR values are represented in decibels (dB). The equations to calculate MSE and PSNR are given below:

$$MSE = \frac{\sum_{i=1}^{N} |x_i - \widetilde{x}_i|^2}{N} \tag{3.1}$$

$$PSNR = 10log_{10}\frac{x_{max}^2}{MSE} \tag{3.2}$$

Where x_i is the pixel value in original image, $\widetilde{x}_i$ is the value of corresponding pixel in rendered image, N is the total number of pixels in a frame, and x_{max} is the peak pixel value, e.g., 255 in 8 bit image. The higher the PSNR value, the larger the similarity between the inpainted image to the original. The PSNR is calculated for the coloured images in this thesis, for this first the MSE is calculated for all the three (red, green and blue) channels and then its average is computed. Other quality metrics such as *structural similarity index* (*SSIM*) can also be applied to the image inpainting analysis, but PSNR is chosen over SSIM for two reasons: i) the SSIM metric accounts only for luminance values and does not consider texture information (Martanez-Noriega et al., 2012); the SSIM is determined for entire

image and its applicability to evaluate arbitrary shaped regions is not straightforward (Ndjiki-Nya et al., 2010). In contrast, the PSNR can easily be computed locally for only a specific inpainted region.

Since PSNR calculations are based on per pixel error measurements, errors due to pixel projection and interpolation may also contribute to the measured PSNR. Considering this, for a thorough evaluation of the resulting inpainting performance of the new framework, the PSNR is separately calculated for both the whole image and the inpainted region only. For whole image, the PSNR calculation includes all types of errors involved in the view synthesis process, whereas in the inpainted region only case, only the inpainted hole region is selected for PSNR computation against the available ground truth image that targets only the disocclusion hole regions, which is the prime focus in this thesis.

3.6.2 Qualitative Assessment

Evaluating the performance of tested algorithm solely on the basis of quantitative metrics does not characterise the image quality particularly well (Girod, 1991; Tan et al., 2005; Wang et al., 2004). In certain cases it is observed that images with good perceptual quality may have lower PSNR values (Azzari et al., 2010). This is because PSNR does not always capture the distortion as perceived by a human being (Martanez-Rach et al., 2014). Since humans are the end users, methods to assess the visual quality of images by human observer are also important (Seshadrinathan et al., 2010). To strengthen the performance assessment criterion, qualitative image analysis is also performed through visual inspection.

Due to the inherent subjective nature of the inpainting process alongside the objective metrics, perceptual assessment helps in analysing results of both texture image and depth maps. The texture image inpainting results are visually examined by comparing them against the available ground truths and selected comparators to test the performance. The key focus areas for visual inspection are the holes that appear around the object boundaries. The inpainted hole regions are compared with the corresponding available ground truth image by observing any artefacts such as unnatural object borders, visual annoyance due to merging of FG objects into the BG or improper filling as compared to the ground truth etc. (Azzari et al., 2010). During the results discussion in Chapters 4, 5, 6 to highlight the problem regions, a subset of inpainted images are enlarged and compared against the subset of ground truth images and the corresponding comparators. However the inpainted depth maps cannot be compared directly against a depth ground truth due to the unavailability of original depth image. Thus the evaluation of inpainted depth map is performed by visually inspecting for smooth inpainted edges around object boundaries and comparing with the texture image to detect the object boundaries. This helps in determining the object shapes and thus detecting the corresponding depth variations based on the FG and BG depth.

3.6.3 Inpainting Time Analysis

Additionally in this thesis, inpainting time has been discussed as a performance metric for disocclusion inpainting. Since inpainting is an iterative process, the PSNR vs time analysis evaluation was undertaken on per iteration basis, with log files being maintained. The total computation time incurred per iteration for

different algorithms has been discussed in detail in the contribution Chapters 4, 5 and 6. At various stages wherever appropriate, time complexity analysis have been carried out amongst different patch size and with other comparators. In Chapter 4, the quantitative and qualitative evaluation is performed by repeating the experiments for different patch size ($w \times w$) varying from 5×5 to 13×13 pixels. These were chosen to investigate their impact on the inpainting performance in terms of qualitative, quantitative and time complexity. This investigation assisted in selecting an appropriate patch size for the proceeding contribution chapters. Chapter 6 employs pMATLAB to improve the inpainting time.

3.7 Software Code Validation

Code validation is significant in assuring the reliability of implemented software. To validate the implemented code both static analysis checks and dynamic tests have been undertaken. Static Code analysis identified and diagnosed run-time errors such as overflows, divide by zero whereas dynamic tests included unit-tests to independently scrutinise each testable part for proper operation.

Furthermore, MATLAB Profiler function was employed to improve the run time performance of the code. The error detection during profiling helped in isolating the problem and thus troubleshooting. Some functions deployed in the inpainting framework were available as MATLAB functions in its image processing toolbox (Mathworks, nd), while other functions are publicly available for direct implementation (Bhat, nd; Wilmer, 2003).

Iteration no.	**Time (sec)**	**PSNR (dB)**	**Target row**	**Target column**	**Candidate row**	**Candidate column**	**Error value**
1	2.1061	19.2586	196	99	69	98	1525
2	2.2288	19.2789	200	100	157	107	3759
3	2.2822	19.2981	204	101	161	108	4881
4	2.2174	19.3	147	340	331	420	12529
5	2.0916	19.3495	181	95	106	83	4885
6	2.0862	19.3839	242	111	242	116	2708
7	2.3146	19.3989	208	102	82	123	10628
8	2.2523	19.4191	246	112	246	117	4776
9	2.1509	19.4226	169	436	167	346	13802
10	2.7748	19.4232	169	440	68	431	14879

Table 3.2: An extract from a test case log file.

1. The correct behaviour of the code implementation of inpainting framework and comparator was manually checked by using a number of test cases.

2. To validate the iterative inpainting process, log files were generated. An extract from test case log file is provided in Table 3.2, which shows target and candidate patch indexes, along with their computed error values, iteration time and corresponding PSNR. These log files served three purposes:

   (a) To cross-check the location of selected target patches in the reference to the theoretical calculations. This determined that the priority order for iteratively filling of holes is implemented correctly.

   (b) To detect and verify the location of candidate patches and the associated error value.

(c) To analyse and evaluate the performance of the proposed framework against the comparators by plotting the iterative time graphs for inpainting time analysis.

3. Test functions were used to determine if all holes are filled in the final synthesised image. For example, the missing pixels were represented as *NaN* (Not a Number) initially and the *RGB* values were assigned to them during each iteration step. In the end, a MATLAB function (*e.g. isnan*) was used to identify if all the holes have been filled.

Further, each constituent component of the implementation was also tested independently for its functionality. Details of the individual comparators used for validation and results evalution will be specified in respective contribution chapters.

3.8 Summary

This chapter has presented the research methodology adopted in this thesis. A detailed description of the test-bed used to implement the inpainting framework for different scenarios has been discussed. The choice of image datasets for experimentation has been justified and the performance metrics selected to evaluate the final synthesised view have been identified. Various benefits underlying selection of MATLAB platform have been highlighted. Details of rigorous testing and validation methods for assessing the inpainting framework have been presented. The next chapter will present the first contribution, namely *Joint Texture-Depth Inpainting*.

Chapter 4

Joint Texture-Depth Inpainting

4.1 Introduction

As discussed in Chapter 2, transmitting texture and depth maps from one or more reference views enables a user to freely choose virtual viewpoints which are synthesised via DIBR. In each DIBR-synthesised image, however, there remain disocclusion holes with missing pixels that correspond to spatial regions occluded from reference view images. To complete these holes, previous schemes (Daribo and Pesquet-Popescu, 2010; Gautier et al., 2011; Meur et al., 2011; Oh et al., 2009) rely on the availability of a high-quality depth map in the virtual view for inpainting of corresponding texture map.

The underlying assumption for the majority of these works however, is that a complete and good-quality depth map at the target virtual view is available, or can be easily pre-computed a priori, for the computation of the priority term

and/or matching criteria. This assumption is not realistic for practical DIBR view synthesis systems; disoccluded pixel locations in the target virtual view with missing texture information will also have depth information missing. Further, though depth maps are known to be piecewise smooth, the missing depth pixels can be more complex than a constant BG depth value, meaning simple signal extrapolation strategies extending the depth signal of the neighbouring BG pixels will not always be correct.

In this chapter, a new inpainting technique called *Joint texture and Depth Inpainting* (JTDI) is proposed to jointly inpaint texture and depth pixels in disoccluded regions, where first available depth information is leveraged to fill in texture pixels, then the inpainted texture information is used to fill missing depth pixels. To facilitate this joint texture and depth filling, an inpainting technique based on Exemplar-Based Inpainting (EBI) (Criminisi et al., 2004) is presented with a new depth-based priority term.

The next section presents the JTDI technique and discusses the steps involved.

4.2 Joint Texture-Depth Inpainting

The EBI for regular colour/texture images has been discussed in Section 2.5.4. The priority computation in EBI involves both a confidence term and a data term but does not utilise depth information. Depth-Assisted Inpainting (DAI) (Daribo and Pesquet-Popescu, 2010) used depth variance term to modify the computation of priority term but it assumes the availability of pre-computed depth map. This

section presents JTDI, which contains two main contributions: 1) a new depth-based priority computation based on EBI to inpaint texture and 2) simultaneous inpainting depth disocclusion holes guided by the inpainted texture. Figure 4.1 shows the detailed block diagram for JTDI and the various steps involved are discussed below:

Step (1): Novel Depth-based Priority Computation

This step computes a depth-based priority term for JTDI. As discussed in Section 2.5.5, disocclusion hole is a spatial region that is occluded by a closer object in the reference view, but become visible in the virtual view. The *a priori* information is that disocclusion areas typically occur at FG object boundaries and these areas are required to be filled with the BG information. Also selecting an appropriate priority order is crucial, as a patch filled from FG boundary initially will lead to serious error propagation in the following iterations and cause FG leakage into large spatial area. Thus, a suitable priority term is required to be computed such that the filling order begins from BG to FG. Although DAI proposed to give higher priority to patches on the BG by selecting the patches with smaller depth variance, it does not assure that the patches are filled starting from BG to FG boundary.

To make sure that BG patches are inpainted first, a new depth-based priority is computed to provide higher priority to patches with smaller depth mean. This is because, the patches farther away from the camera belongs to the BG and have smaller depth values (i.e. $Z_{far} = 0$, $Z_{near} = 255$). Also the depth variance term which is incorporated as a multiplier to the original terms $C(p)$, $D(p)$, $L(p)$ in

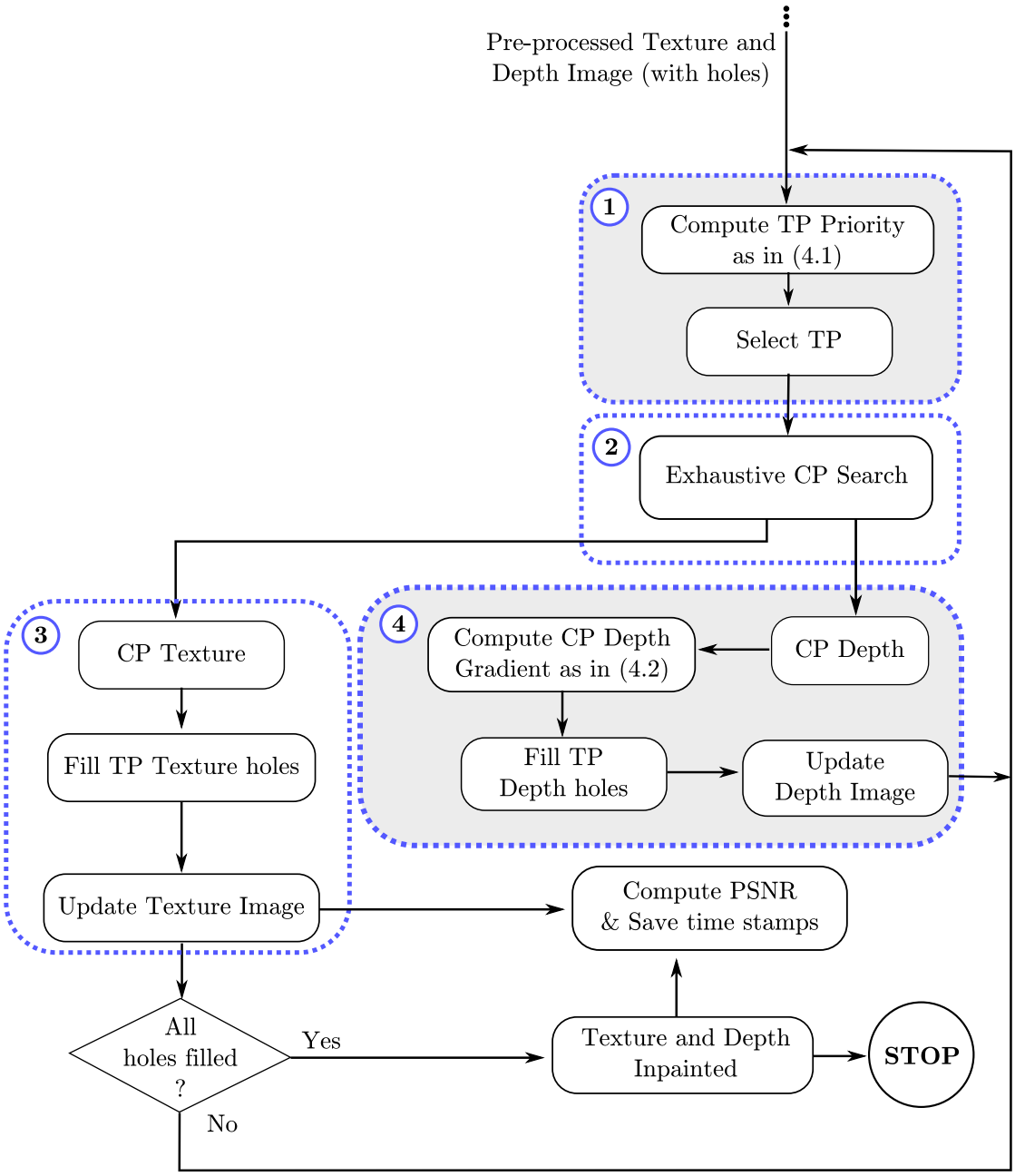


Figure 4.1: JTDI with contributions highlighted in step (1) and (4).

(2.9) are now combined additively instead. The rationale behind adding these terms is to overcome the circumstances where patch priority reduces to zero apart from having high confidence $C(p)$ and low variance $L(p)$ terms. Such a condition occurs when the data term $D(p)$ tends to zero (Nie et al., 2006). The additive combination provides equal weightage to all participating terms. Thus the priority

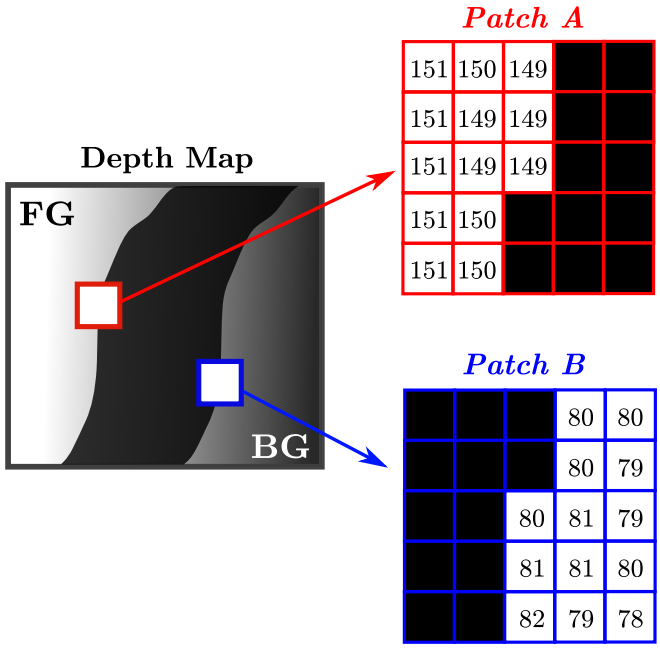


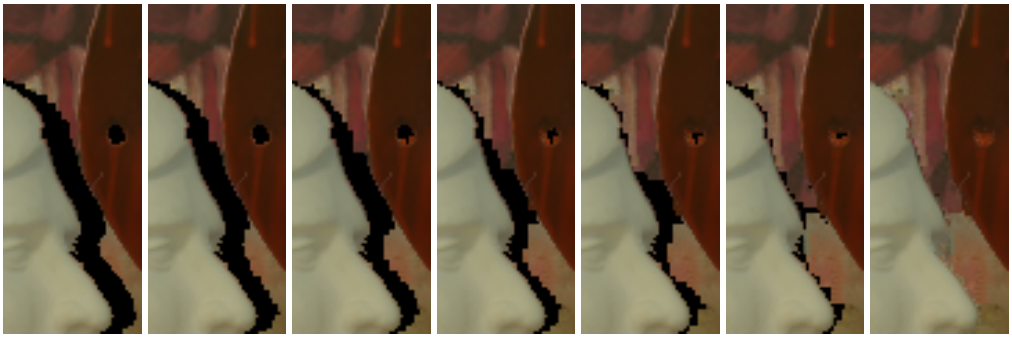
Figure 4.2: Depth map showing per pixel values in patch A (in *red*) and patch B (in *blue*) respectively.

term $P(p)$ in (2.9) is revised as:

$$P(p) = [C(p) + D(p) + L(p)] \times (Z_{near} - \overline{Z}_p) \tag{4.1}$$

where $Z_{near} = 255$. Unlike DAI, the depth mean term is a clear dominant term in the computation of $P(p)$, so that patches further in the BG are always selected for inpainting first. The priority is thus computed for each patch of size $w \times w$ at the holes boundary using (4.1). The overall aim is to select the TP from the BG which is attained by additional depth mean term to the low depth variance in the new priority computation.

A worked example is used to illustrate the improved priority term which aims to perform BG to FG filling. Figure 4.2 shows an example of a depth image with two patches namely *patch A* and *patch B* on FG and BG hole boundary respectively. The patches A and B of size 5×5 each, are zoomed-in to represent their per-pixel



Stage I Stage II Stage III Stage IV Stage V Stage VI Stage VII

Figure 4.3: Inpainting of *Art* at different iterations from Stage I to Stage VII illustrating BG to FG filling order.

values in *red* and *blue* matrices respectively. Assume that both these patches have same confidence and data terms. Using (2.10), $L(p)$ of *patch A* and *patch B* is calculated as 0.71 and 0.64 respectively. This provides higher priority to *patch A* on the FG rather than *patch B* on the BG. However, with the multiplicative mean depth-based priority term in (4.1), *patch B* on the BG has higher priority over *patch A* on the FG. This is because the $(Z_{near} - \overline{Z}_p)$ for *patch A* and *patch B* are 105 and 175, respectively. Thus, it clearly shows that the new priority order with depth mean term supports the BG to FG filling by giving higher priority to patches on the BG.

Figure 4.3 shows another example which illustrates the improved priority order from BG to FG through various stages from Stage I-VII. Using new depth-based priority selection, higher priority is given to the patch farthest in the BG which helps in starting the filling process from the BG boundary. This reduces the chance of selecting a TP at FG object boundary i.e. sculpture face, and thus minimises the propagation of unwanted artefacts at the object boundaries. Stage I shows when no holes are filled, followed by Stage II to VI illustrating the selection of TP on the BG at each consecutive iteration which results in improved final inpainted

output in Stage VII. The selection of TP on the FG at any stage would have led to leakage of FG in to BG and distort the face of the sculpture.

Step (2): Exhaustive Candidate Search

After the TP is selected, the best CP is searched by performing an exhaustive TM in the candidate search space, as in EBI. The candidate search space contains overlapping patches from the known region in the image. The error is computed among the TP and each CP in the search space as in (2.4), the best matched candidate having the lowest error is selected for filling TP.

Step (3): Inpainting Texture Disocclusion Holes

The missing pixels in TP are filled with corresponding known pixels in the selected CP. However, the depth holes still exist and the next step explains the inpainting of target depth holes guided by newly filled TP.

Step (4): Inpainting Depth Disocclusion Holes

The key novelty of JTDI algorithm exists in simultaneous texture and depth inpainting. JTDI alternates between inpainting of texture pixels using partially available depth information, and then inpainting of missing depth pixels using inpainted texture information. Specifically, after the best-matched texture patch $\Psi_{\hat{q}}$ is found in the source region Φ, the corresponding depth patch Z_q is used to fill in missing depth pixels in target depth patch Z_p as follows:

$$Z_{\hat{p}} = \overline{Z}_p - (\overline{Z}_q - Z_{\hat{q}}) \quad where \quad \overline{Z}_p \geq \overline{Z}_q - Z_{\hat{q}} \tag{4.2}$$

otherwise $Z_{\hat{p}} = \overline{Z}_p$. Here, $\overline{Z}_p$ and $\overline{Z}_q$ are the mean depth values of the target depth patch Z_p (computed using available depth pixels) and the best-matched depth patch Z_q, respectively. $Z_{\hat{p}}$ is the missing target depth pixels in Z_p and $Z_{\hat{q}}$ represents its corresponding candidate depth pixel in Z_q. In other words, only the depth gradient of the matched patch Z_q is copied to the target, while the depth mean of the original patch Z_p (based only on initially known pixels in the TP) remains the same.

The rationale for (4.2) is as follows: TM between texture patches just ensures the textural patterns are similar; the patches could be from quite different depths of the 3D scene, e.g. same wallpaper pattern recurring on a wall slanted towards infinity away from the camera. Thus, directly copying of depth pixels from best-matched patch (evaluated solely on texture content) to the TP, is a tenuous proposition. On the other hand, given the textural content are similar, the depth gradient of the best matched patch is more likely to be similar to the gradient of the TP, as illustrated in the aforementioned wallpaper example. Thus copying only the depth gradient to the target depth patch is more appropriate. Finally, by retaining the original mean depth value Z_p in the TP, a piecewise smooth depth map can be achieved.

The outputs of step (3) and (4) provide inpainted texture and depth holes in the TP. The inpainted holes are updated in the synthesised image and priority is computed for next iteration. The steps (1) to (4) are repeated in an iterative

manner until all the holes are filled. The final output image after all the holes are filled is an inpainted virtual view comprising of both texture and depth map. The next section critically evaluates and discusses the experimental set-up and results for JTDI.

4.3 Experimental Set-up and Results

Inpainting experiments are performed on *eight* Middlebury image datasets to investigate and analyse the JTDI technique. This section provides a detailed discussion on *four* of these datasets and results for other datasets are included in the Appendices B and E. To test the performance of the designed algorithm, results are compared in two scenarios: *Experiment 1* presents results of inpainting performed on views synthesised via DS-DIBR and are compared against MVSV (Ramachandran and Rupp, 2012) whose experimental results are available for Middlebury datasets. *Experiment 2* presents the results for views synthesised through SS-DIBR and compared against EBI and DAI. EBI is a pioneer work in the field of image inpainting, mainly focussing on texture image inpainting while DAI utilises depth information to inpaint missing texture holes. The algorithm for EBI is publically available for direct implementation (Bhat, nd) and the results for DAI are reproduced by employing depth-based modifications to EBI as discussed in (2.9) and (2.10) in Chapter 2. DAI assumes the availability of a complete depth map *a priori*, which is achieved by horizontally extrapolating the BG depth values into the hole region (Xu et al., 2013). Both the scenarios are discussed and presented in Section 4.3.1 and 4.3.2, respectively.

4.3.1 *Experiment 1*: Inpainting DS-DIBR Views

The experimental results of proposed JTDI are compared for *three* different virtual views generated at V2, V3 and V4 using V1 and V5 (i.e. $\alpha = 0.25$, 0.5 and 0.75, as discussed in Section 3.3). The inpainting results for 4 datasets have been compared with MVSV(Ramachandran and Rupp, 2012) for quantitative performance analysis. The PSNR is computed for whole images and the PSNR comparison of JTDI and MVSV is presented in Figure 4.4 which clearly shows that JTDI performs considerably better then MVSV.

For example, a modest improvement is observed in the *Aloe* dataset, where an average PSNR increment is 4.23 *dB*. However, for the *Art* dataset, the PSNR rise is 0.30 *dB*. This variation in improvement indicates that during the candidate search, *Aloe* has found enough good CPs whereas in *Art,* there remained scarcity of good candidate matches with lower MSE and thus resulted in overall lesser PSNR improvement in comparison to other datasets. It is evident from plots that JTDI holds an upper edge over its comparator due to significant increase in PSNR for majority of datasets synthesised at V2, V3, and V4, respectively.

The overall average plot for these datasets is shown in Figure 4.5 and it is observed that JTDI outperforms MVSV. The comparisons for more datasets and their summary is presented in Appendix B (Figure B.1 - B.3) for completeness.

During this experimentation, it emerged that due to in-built image characteristics and unavailability of good candidate patches, certain image datasets encountered less improvement over other datasets. However the qualitative analysis could not be performed since the perceptual results of MVSV are unavailable in

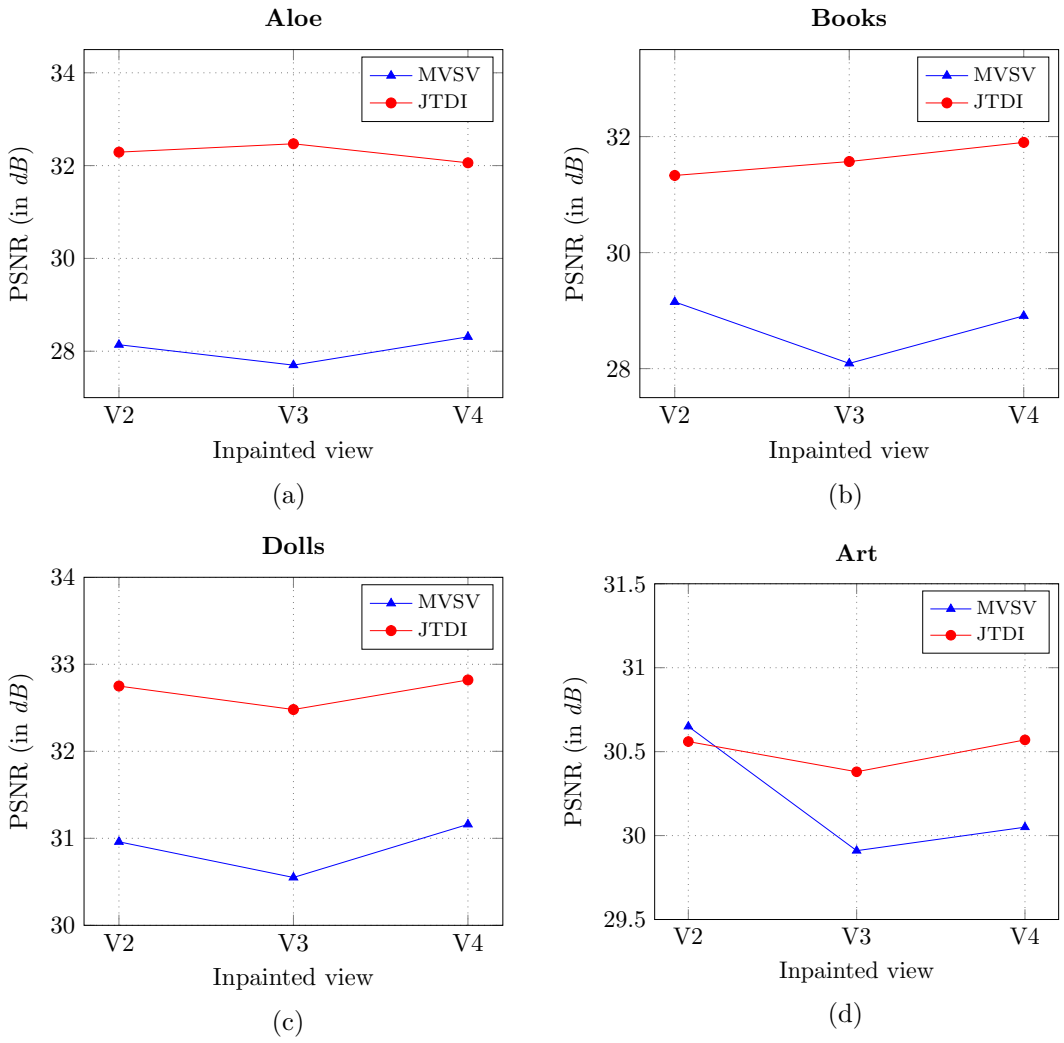


Figure 4.4: PSNR comparison of MVSV and JTDI for (a) *Aloe* (b) *Books* (c) *Dolls* and (d) *Art,* for three views (V2, V3 and V4).

public domain. Experiment 1 deals with inpainting of small disocclusion holes as these experiments are conducted on the view synthesised using DS-DIBR.

In order to test the robustness of the algorithm, Experiment 2 is performed for inpainting of larger hole regions.

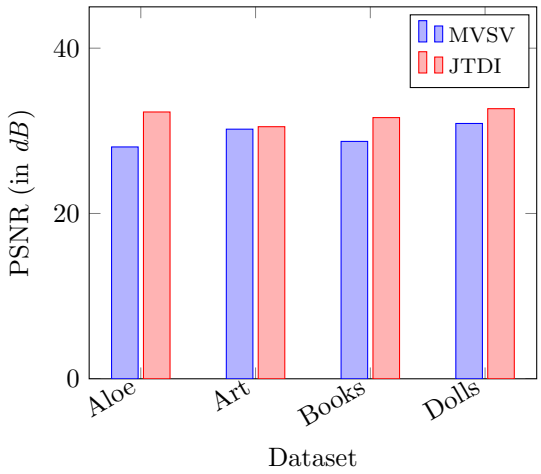


Figure 4.5: Average PSNR comparisons for MVSV and JTDI.

4.3.2 *Experiment 2*: Inpainting SS-DIBR Views

In this experiment, SS-DIBR is used to generate V3 (view #3) using reference V1 (view #1) for *Aloe* and V2 (view #2) to generate V4 (view #4) for *Cones*. The view generation using SS-DIBR poses a bigger challenge for JTDI and amounts to increased number of holes for disocclusion inpainting. The disocclusion holes appeared in synthesised texture and depth maps are simultaneously filled using JTDI.

The results of the implemented JTDI for 4 datasets namely: *Aloe*, *Art*, *Cones* and *Laundry* are evaluated and compared both quantitatively and qualitatively against state-of-art EBI and DAI in the next section.

4.3.2.1 Quantitative Analysis

To test the quantitative performance, PSNR is computed for both 1) Whole Image and 2) Inpainted Region as discussed in Section 3.6.1. The comparative analysis for JTDI, EBI and DAI has been performed for 5 different patch size i.e. $w = 5, 7, 9,$

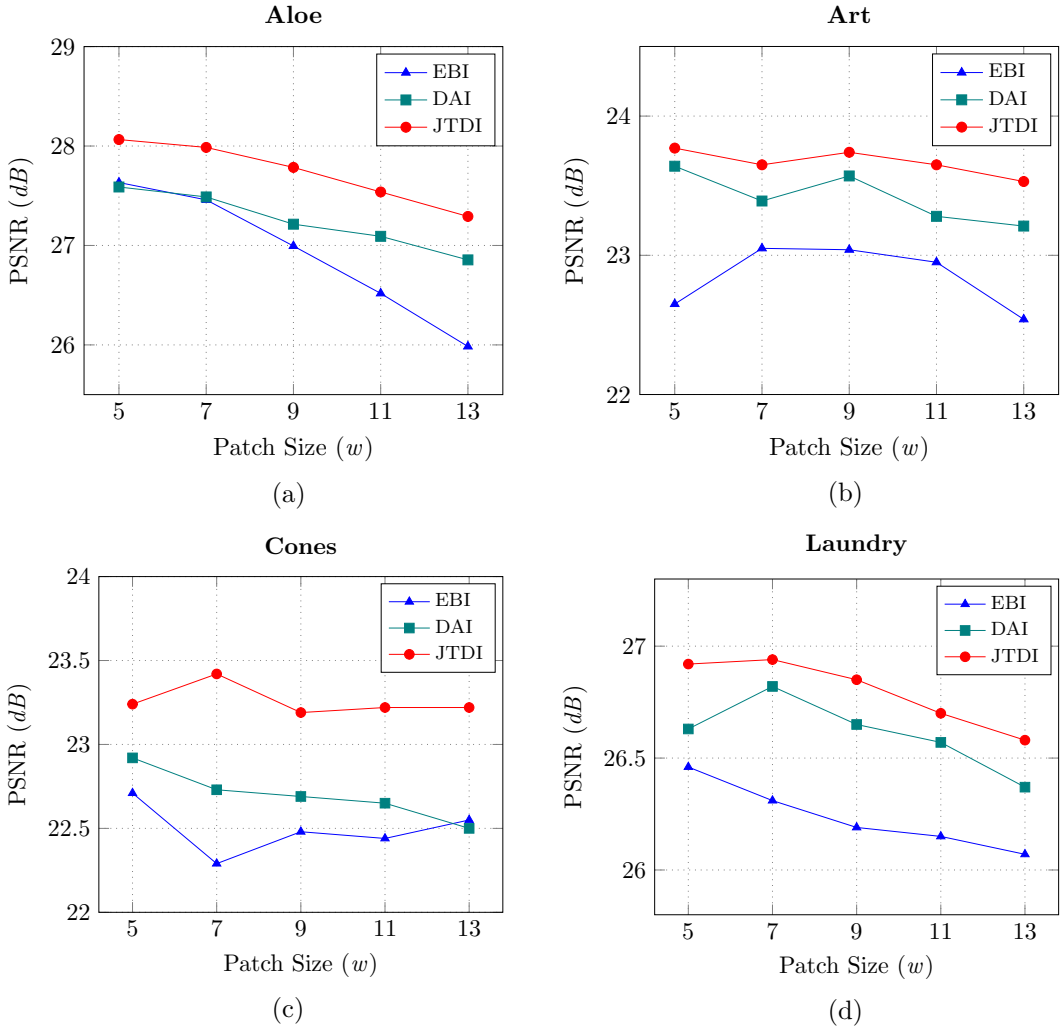


Figure 4.6: Whole image PSNR vs patch size (w) plots of EBI, DAI and JTDI for (a) *Aloe* (b) *Art* (c) *Cones* and (d) *Laundry*.

11, 13, to evaluate the effect of w on the overall inpainting performance. The plots in Figures 4.6 (a) - (d) show the PSNR vs. patch size performance comparison of *Aloe, Art, Cones* and *Laundry,* for EBI, DAI and JTDI, respectively. These results demonstrate that JTDI performed better than EBI and DAI for all datasets. For example, considering the plot for *Aloe* image in Figure 4.6 (a), best output PSNR is observed at $w = 5$. As w increases, although the PSNR drops for JTDI but it still remains higher as compared to EBI and DAI. Similar performance is consistently maintained for almost all the datasets irrespective of the w involved. Overall the

results suggest that for any given w, JTDI consistently performs better among the comparator inpainting methods.

On comparing the performance of JTDI at different values of w, it is observed that *Art* performs similar at $w = 5$, 7 and 9 e.g. PSNR difference of $w = 9$ and $w = 5$ is 0.25 dB for whole image. However, for *Cones*, the PSNR remains similar at all values of w, except with a increase of 0.18 dB at $w = 7$ compared to $w = 5$. For *Aloe* and *Laundry*, the PSNR decreases as w increases from 5 to 13. It is observed that there is no clear evidence of the single best w which provides the best output PSNR for all the inpainted datasets. However, w has a direct effect on the inpainting time and will be discussed in Section 4.3.2.4.

Figure 4.7 shows the bar plots representing PSNR comparison of inpainted region for all three approaches. For *Aloe* at $w = 9$, the resulting PSNR increases by 8.72% and 6.23% as compared to EBI and DAI, respectively. At same w for *Cones*, it increases to 11.38% and 8.94% in comparison to EBI and DAI. Similar results have been observed for other image datasets, which are shown in Appendix B (Figure B.4 - B.5).

These results highlight that the depth-based BG first priority order helped in sequencing the iterative inpainting process starting from the BG region and moving inwards towards the FG hole boundary. This led to lower error propagation artefacts which rises if the initial filling is performed starting from the FG holes boundary.

The next section presents the results for qualitative analysis.

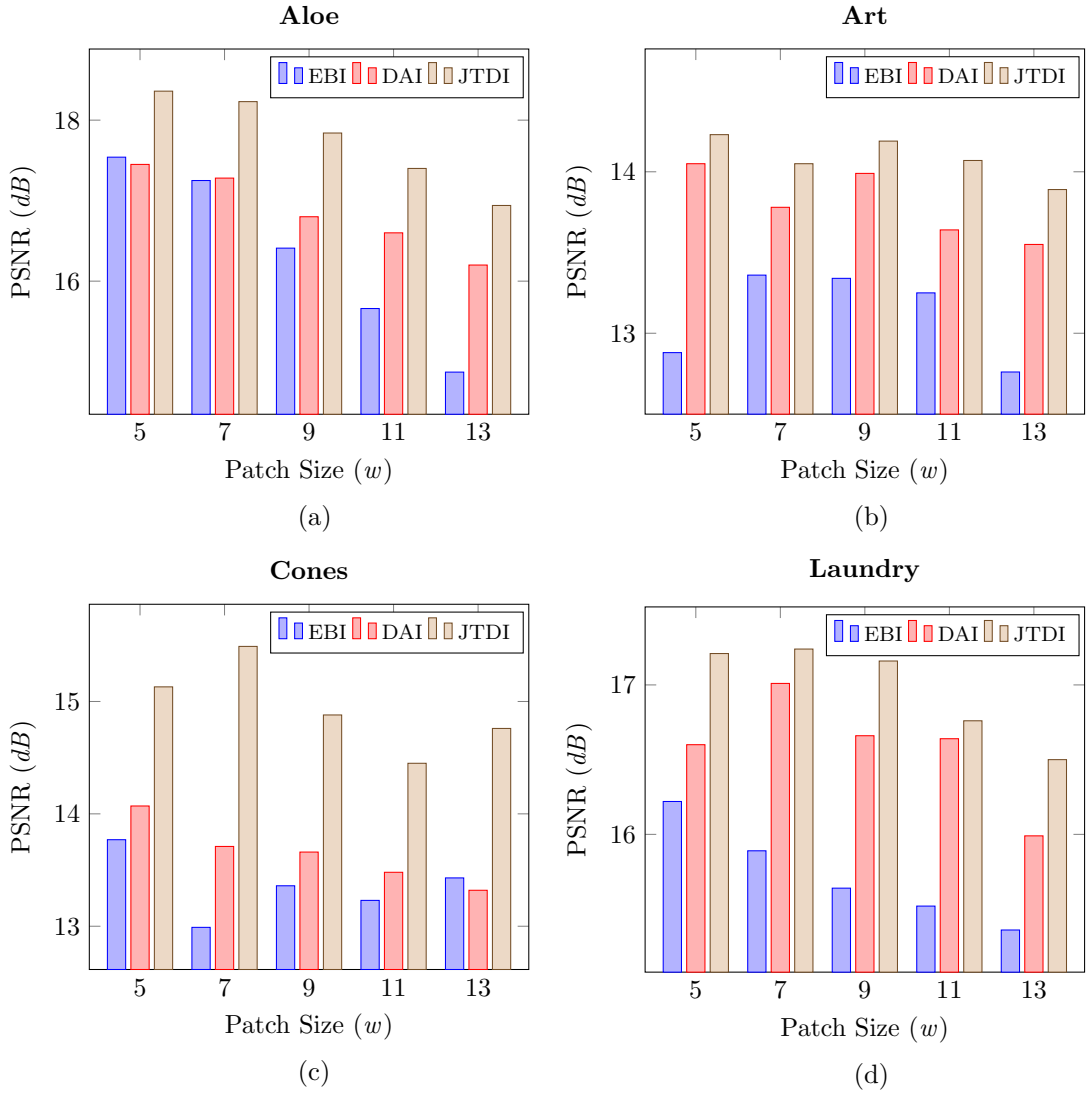


Figure 4.7: Inpainted region PSNR vs patch size (w) plots of EBI, DAI and JTDI for (a) *Aloe* (b) *Art* (c) *Cones* and (d) *Laundry*.

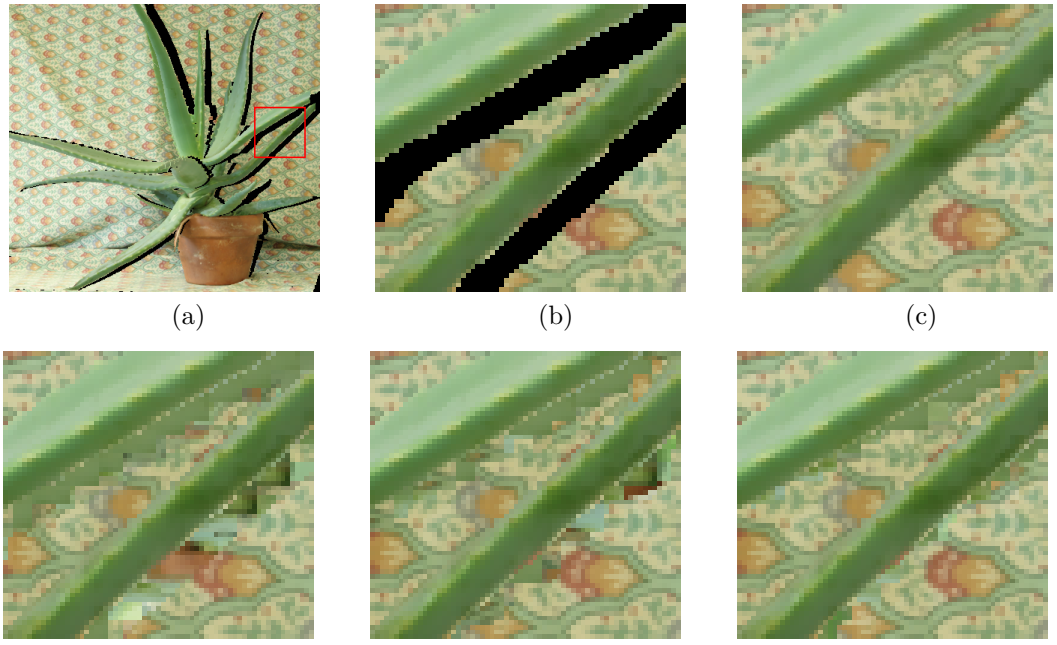
4.3.2.2 Qualitative Results

This sub-section discusses the qualitative performance comparison of JTDI with EBI and DAI for $w = 9$. Figures 4.8 shows the qualitative results for *Aloe*. The area highlighted as *red* in Figure 4.8 (a) is the subset hole region chosen for analysis. This region contains large disocclusion holes and is selected to assist in close visual inspection of the inpainting performance. The sub-region in Figures, 4.8 (b) and (c) shows a zoomed-in view of hole region and ground truth. The corresponding

sub-regions shown in Figures, 4.8 (d) - (f) represent the inpainting results for EBI, DAI and JTDI respectively.

It is observed that the inpainting performance of JTDI around the boundary of leaf is much smoother in comparison to EBI and DAI which exhibit visible artefacts. The performance comparison is based on visual analysis of inpainted holes with respect to the ground truth. On closer inspection of the representative sub-regions in Figures 4.8 (d)-(f), it is observed that JTDI performs better in preserving the FG object boundaries and provides better inpainting results over both the comparators. The reduced artefacts are the result of proposed improved priority term, where the filling process begins from BG and move inwards toward FG as seen in Figure 4.8 (f). Some more examples are shown in Figures 4.9, 4.10 and 4.11 representing results for *Art*, *Cones* and *Laundry* datasets, respectively.

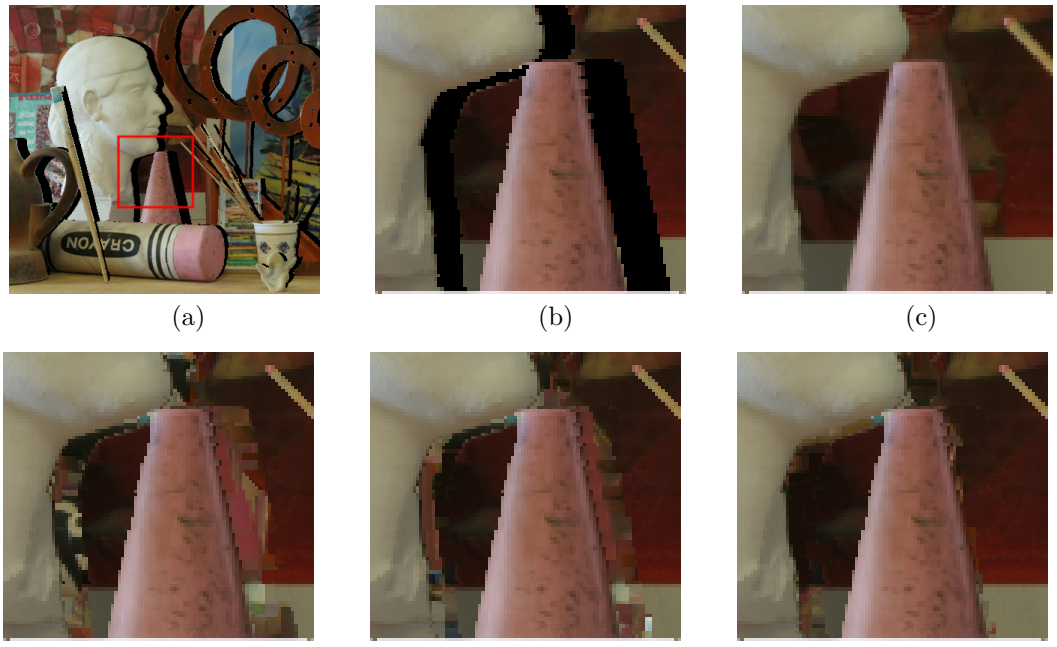
In all these Figures, comparing the inpainting results of EBI, DAI and JTDI amongst themselves and against the respective ground truths, signify the improved performance of JTDI over DAI and EBI. The selection of TP on the FG first, tends to find the best matching CP from the FG regions because the CP is chosen based on the known information contained in the TP. Using this CP to fill the missing holes results in FG leakage into the BG regions as evident in Figures, 4.8(d) - 4.11(d). DAI performs better then EBI due to the depth variance based priority term.



(a) (b) (c)

(d) (e) (f)

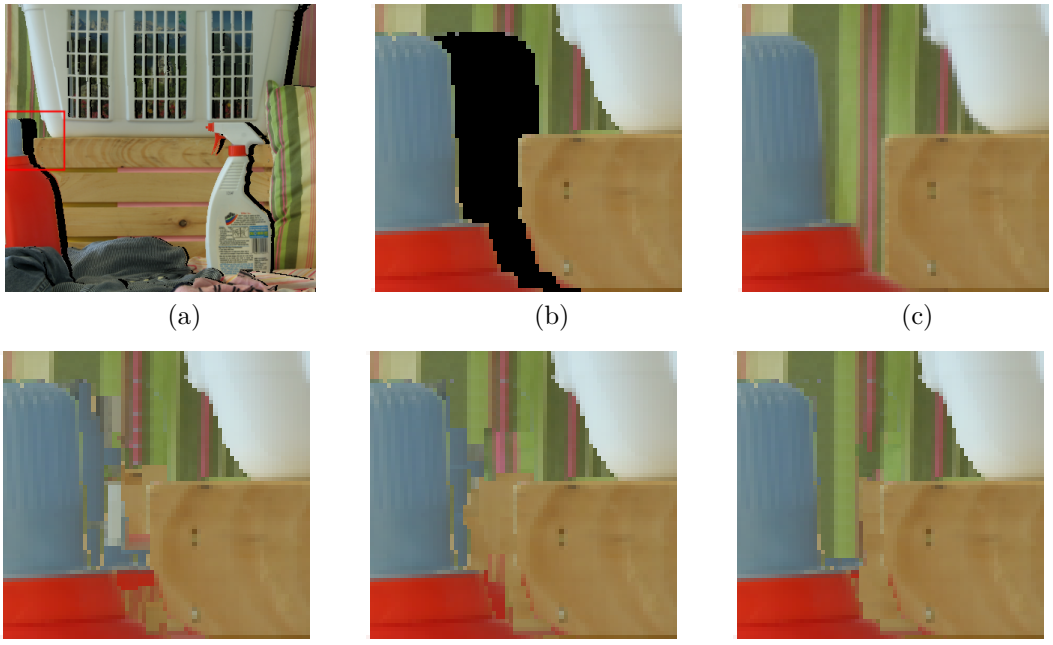
Figure 4.8: Inpainting results for *Aloe* at $w = 9$ with (a) Image with holes (b) Hole sub-region (c) Ground Truth and (d), (e) and (f) represent corresponding inpainting results by EBI, DAI and JTDI respectively.



(a) (b) (c)

(d) (e) (f)

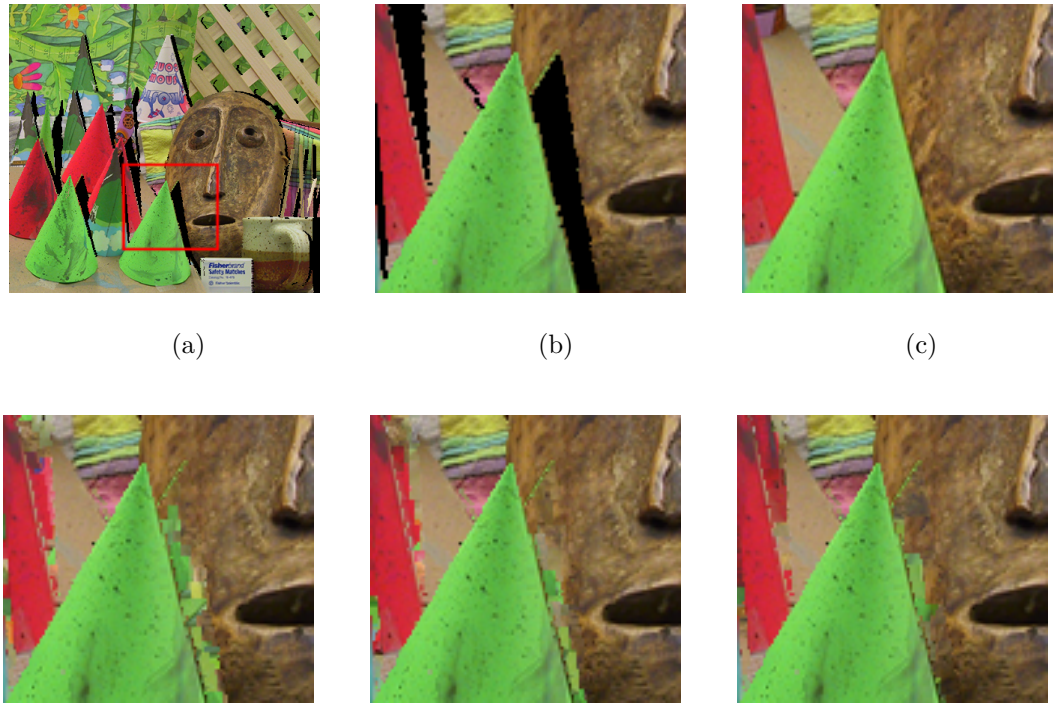
Figure 4.9: Inpainting results for *Art* at $w = 9$ with (a) Image with holes (b) Hole sub-region (c) Ground Truth and (d), (e) and (f) represent corresponding inpainting results by EBI, DAI and JTDI respectively.



(a) (b) (c)

(d) (e) (f)

Figure 4.10: Inpainting results for *Laundry* at $w = 9$ with (a) Image with holes (b) Hole subregion (c) Ground Truth and (d), (e) and (f) represent corresponding inpainting results by EBI, DAI and JTDI respectively.



(a) (b) (c)

(d) (e) (f)

Figure 4.11: Inpainting results for *Cones* at $w = 9$ with (a) Image with holes (b) Hole subregion (c) Ground Truth and (d), (e) and (f) represent corresponding inpainting results by EBI, DAI and JTDI respectively.

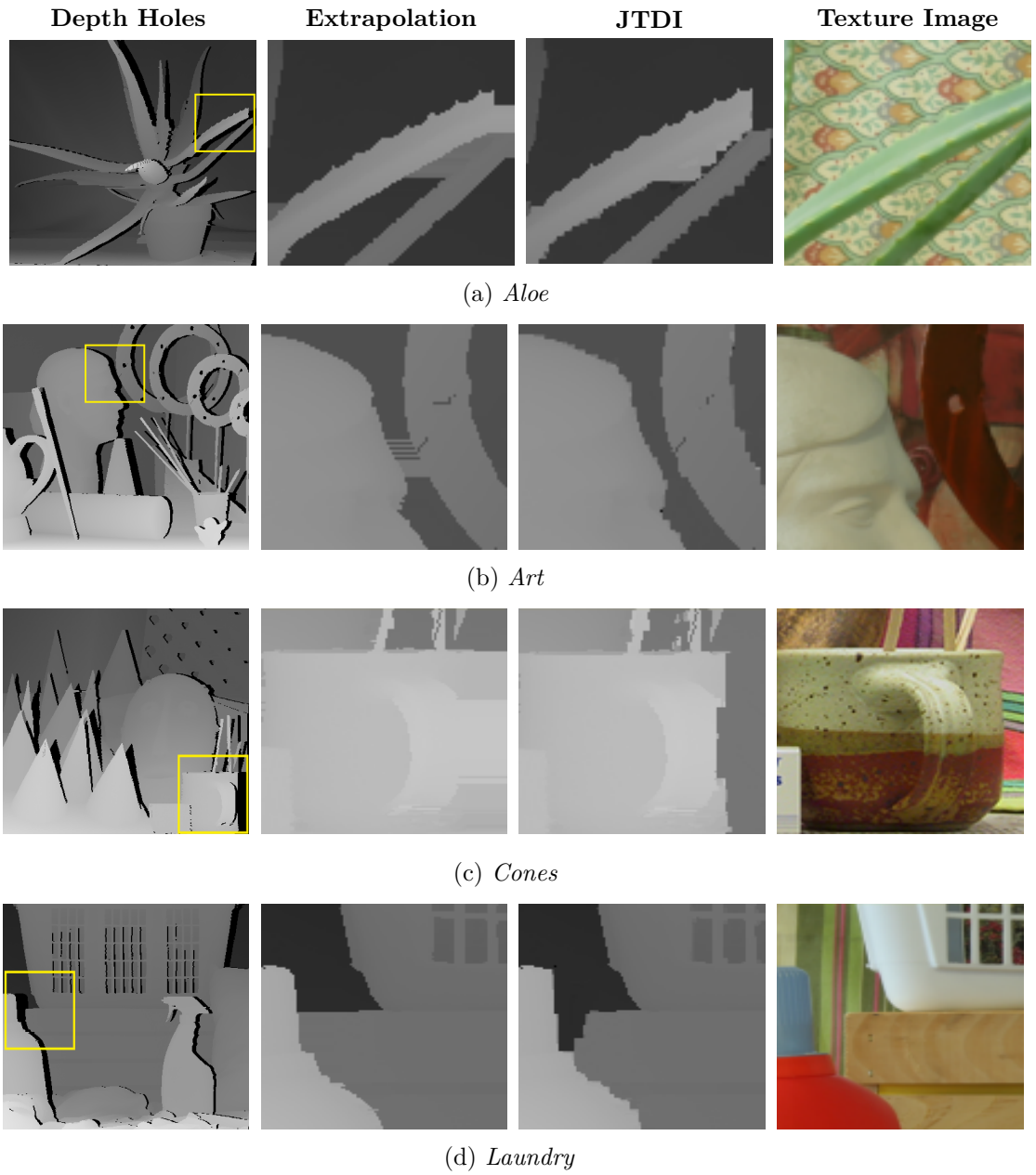
However, the variance term does not always select the TP from BG and hence may introduces the error by selecting TP from FG. The JTDI improved the priority computation and shows better results in comparison to DAI as seen in Figures 4.8(e) and (f) - 4.11 (e) and (f) respectively. Similar trend is seen throughout the experimentation for all the datasets, as presented in Appendix E. The improved filling in JTDI for all these datasets supports the fact that the new priority order provides a superior and robust inpainting, in comparison to the other comparators.

4.3.2.3 Depth Inpainting Results

The comparative results for depth disocclusion hole-filling are shown in Figure 4.12. Due to unavailability of ground truth, the depth inpainting results are compared mutually. Here, the corresponding texture images are included, to detect the object boundaries which reflect the depth boundaries, for performance examination. The inpainting at object boundaries by extrapolation in DAI (column 2) and JTDI (column 3) are compared against the texture images in column 4.

In Figures 4.12 (a) - (d), it is observed that the depth is inpainted better by preserving the object boundaries in column 3 as compared to column 2. The extrapolation method, fills the lowest depth value across the holes into entire region which tends to propagate the FG depth information e.g. if the holes occur between two FG objects as seen in *Art* column 2 Figure 4.12 (b). A similar observation is made for all the datasets, which shows that depth inpainting results obtained via JTDI are better compared to pre-depth filling via extrapolation.

The depth inpainting results for more datasets are included in Appendix E, for



(a) *Aloe*

(b) *Art*

(c) *Cones*

(d) *Laundry*

Figure 4.12: Depth inpainting results for (a) *Aloe*, (b) *Art*, (c) *Laundry* and (d) *Cones*. Column 1 & 4 show the depth map holes and the corresponding ground truth texture image, column 2 & 3 represent inpainting results using extrapolation and JTDI respectively.

completeness. Overall, it is evident, that the simultaneous inpainting of texture and depth maps result in better inpainting of both views and improve the visual quality.

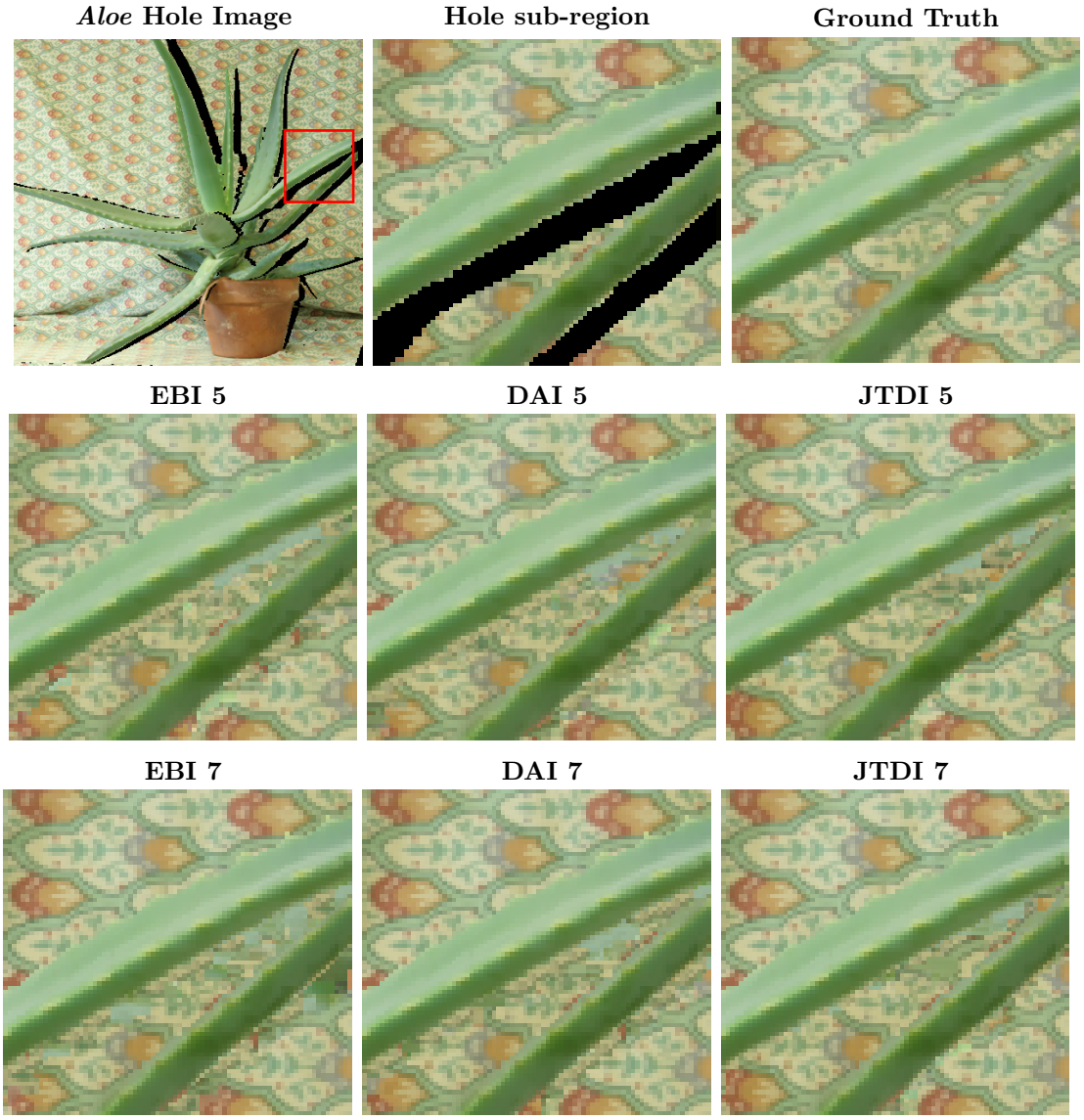


Figure 4.13: *Aloe* inpainting results for EBI, DAI and JTDI at $w = 5$ and 7, with sub-region indicated as *red* box.

4.3.2.4 Patch Size vs Inpainting Time Analysis

As discussed above in Experiment 2, JTDI consistently performs better at given w values for different datasets. To analyse the impact of w on the inpainting time, Figure 4.13 and Figure 4.14 show visual inpainting results for *Aloe* dataset at $w = 5$, 7 and $w = 9$, 11, 13, respectively. Row 1 in Figure 4.13 shows *Aloe* hole image which highlights a sub-region which is zoomed-in for visual inspection and also

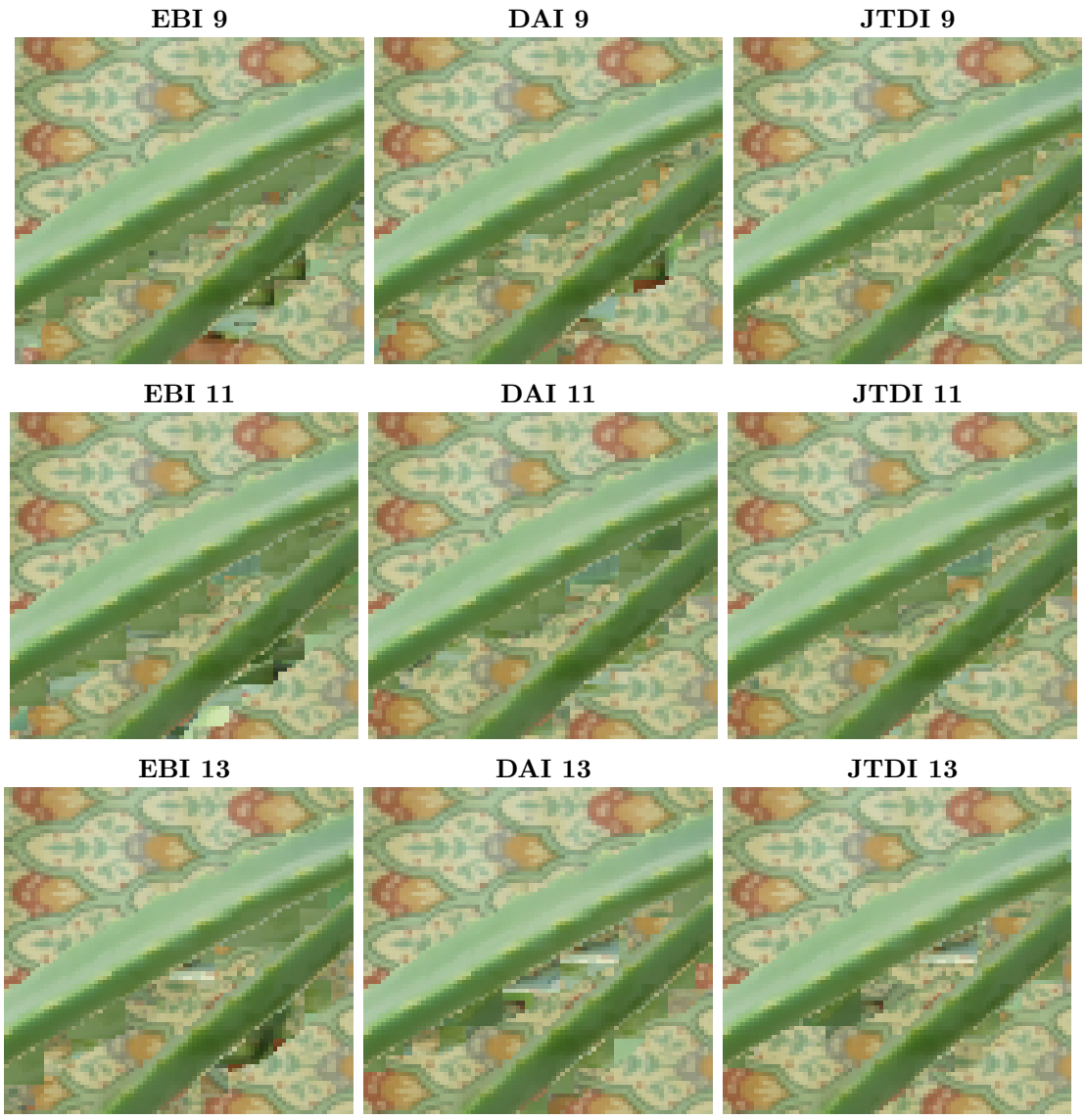


Figure 4.14: *Aloe* inpainting results for EBI, DAI and JTDI for $w = 9$, 11, and 13 respectively.

shows its corresponding ground truth. Row 2 and row 3 represent the inpainting results by EBI, DAI and JTDI at $w = 5$ and 7, respectively. In continuation, row 1, 2 and 3 in Figure 4.14 show corresponding results of EBI, DAI and JTDI at $w = 9$, 11 and 13. On inspection, the best visual quality is observed at $w = 5$ for all three comparators, however, JTDI provides overall best inpainting compared to EBI and DAI, at any given w.

It is observed that as w increases from 5 to 13, the inpainting quality degrades

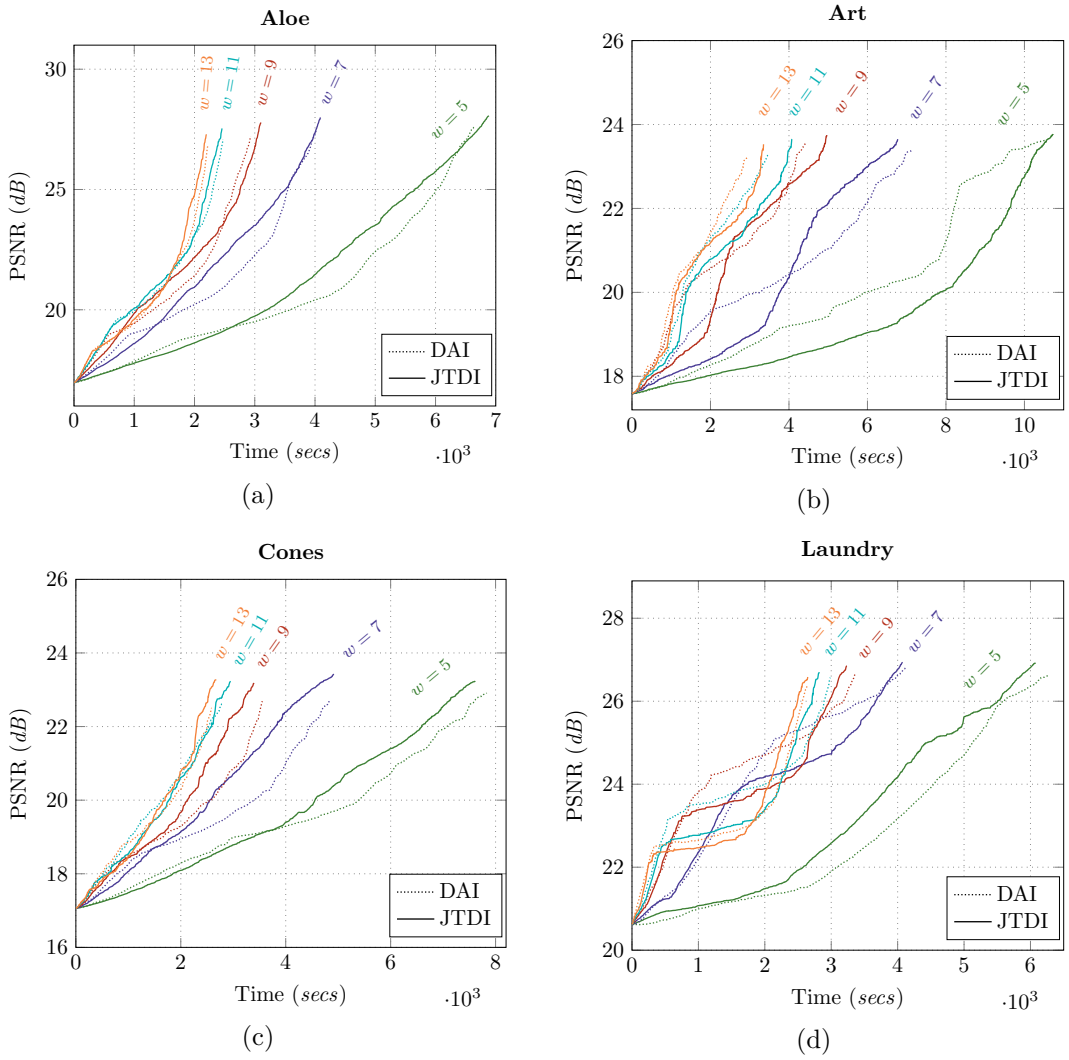


Figure 4.15: PSNR vs Time plots for DAI and JTDI at $w = 5$, 7, 9, 11 and 13 for (a) *Aloe*, (b) *Art*, (c) *Cones* and (d) *Laundry* respectively.

for all the comparators but the corresponding inpainting time involved in inpainting process decreases. This relation between the w and inpainting time is shown in Figure 4.15 for *Aloe*, *Art*, *Cones* and *Laundry* datasets. The iterative PSNR results have been plotted for JTDI and DAI for all w. EBI has not been included in these plots since it does not employ depth information during the inpainting process and is an entirely texture filling method.

The overall inpainting time involved in both JTDI and DAI remains almost similar for all w. The variation in time is related to the priority computation

which affects the number of iterations required to fill the holes for a given w.

As evident from the plots, the inpainting time is in inverse relation to w and it is observed that for a significant number of datasets, smaller w improves the inpainting quality but at the cost of high inpainting time. As w increases from 5 to 13, inpainting time decreases drastically but with the corresponding decrease in the output PSNR. Thus, for an efficient inpainting performance, a trade-off is required between w and the inpainting time.

To select an appropriate w for inpainting, on reference to *Aloe* Figure 4.15 (a), as w is increased from 5 to 7, the output PSNR decreases by 0.08 dB but the total time decreases by almost 40%. Subsequently, when w is increased to 9, it is observed that the inpainting time declines by 55 % with a small reduction of 0.20 dB in PSNR. But as w is increased further from 9 to 11 and 13, the time decreases to 79% but the corresponding inpainting performance and PSNR eventually degrades by 0.80 dB.

Thus, from the plots, for a balanced trade-off between inpainting performance and time, $w = 9$ is considered to be most reasonable choice. Although the output PSNR is slightly smaller at $w = 9$ as compared to $w = 5$, the corresponding inpainting time reduces drastically. This negligible decline in PSNR does not necessarily result in low perceptual quality but it reduces the time by almost 55%. This w analysis facilitates in achieving better inpainting quality but finding w is not the primary aim of this thesis.

4.4 Summary

In this chapter, a new inpainting technique is proposed which simultaneously fills the missing pixels in both texture and depth maps. In particular, using partial depth information a new priority term is defined to order pixel patches in the disocclusion region to be inpainted. Then for a given best matched patch in the source region, the depth gradient of the best-matched patch is copied to the TP for depth inpainting. JTDI is a robust inpainting technique as it can tackle the more realistic DIBR view synthesis scenario where both the texture and depth pixels in the disoccluded regions are missing and challenging to complete. Experimental results show that the proposed mutual assistance inpainting approach has noticeable performance gain in both quantitative and qualitative over other methods.

From the above discussion this is evident that JTDI performs well for majority of image datasets but still there exists a scope for improvement. Certain regions near the object edges are still unsmooth and contain artefacts. It is observed, these artefacts tend to occur as a result of an exhaustive search process adopted during TM; such that the selected CP used to fill the missing region happens to belong to the FG due to its proximity to the TP in terms of MSE. This filling from the FG then results in propagating error boundaries. Improved search methods need to be explored to ensure filling from BG. Another drawback of the exhaustive search scheme is high error computation time during TM. Instead of full exhaustive search, a more focussed approach needs to be investigated which reduces the search complexity with no or minimal loss in inpainting performance.

Another reason for improper filling is the scarcity of best matching candidate patches for a given TP during TM. Such a case appears, if there are inadequate good patches which provide low MSE while TM. It is observed that a small variation among the patches e.g. due to image transformational properties; the resulting PSNR is high and thus a potential candidate patch is dropped out from the TM. To overcome such scenario, Chapter 5 discusses the self-similarity characteristics of image that can be employed to achieve better inpainting.

Chapter 5

Self-similarity Characterisation based JTDI

5.1 Introduction

As discussed in previous chapter, JTDI consistently performed better in comparison to existing methods considered. The performance can be further improved in terms of the visual quality, numerical performance and inpainting time involved. During TM, JTDI searches the best matching CP by identifying similar pixel patches through an exhaustive non-local search i.e. whole image is traversed for TM (Arias et al., 2009). Thus, there exist two main reasons that produce artefacts: firstly, the insufficient good candidate patches lead to selection of inferior CP for inpainting disocclusion holes. Secondly, the exhaustive search process tends to select CP from the FG which results in leaking of FG information into the BG regions. In addition, the exhaustive search methods are computationally expensive

and thus results in higher inpainting time. This chapter aims to address the two aforementioned problems in previous non-local TM schemes.

The natural images tend to possess self-similarity, i.e. similar pixel content appearing repetitively within the image (Ashikhmin, 2001; Fedorov et al., 2016; Lan et al., 2010). However, the similar patches may appear slightly transformed (e.g. scaled or rotated) either due to varying depth or change in viewpoint etc. These transformed patches although being visually similar, results in high MSE during TM and generally overlooked during CP selection. It motivates to investigate and employ these transformed patches as potential candidates during TM. This can be achieved by characterising the self-similarity to detect the transformation parameters for the self-similar patches and then utilise them to enhance the search space for TM. Searching all the transformations of self-similarity at once results in high dimensionality (Barnes et al., 2010; Mansfield et al., 2011) so this chapter exploits the most commonly occurring self-similarity characteristic in images which is *scaling*.

This chapter presents a new *Self-similarity Characterisation* based JTDI (SC-JTDI) to investigate its performance for disocclusion inpainting. Also, the exhaustive search problem is minimised by constraining the search-space only to BG region using depth information which aims to avoid the selection of CP from FG and restrict the FG leaking.

The notion of Self-similarity Characterisation is formally defined in the next section.

5.2 Self-similarity Characterisation

It is observed that natural images are self-similar in general; i.e., a given pixel patch is likely to recur one or more times in non-local spatial regions in the same image. Specifically, the self-similarity is defined as non-local recurrences of pixel patches within the same image - one such characterisation of self-similarity in a given image is across different scales in which these patch recurrences take place. The self-similarity is redefined in a multi-scale manner for natural images: a characterisation of self-similarity for a given natural image is then how well target pixel patches will match with non-local patches of the same image resized by a specified scaling factors.

JTDI inpaint the holes using TM and assume the recurrence of pixel patches in the same scale. In this chapter, the notion is generalised to assume that the recurrence of a pixel patch can take place across multiple scales. Thus, self-similarity is characterised as the *scale parameters* (SP) over which, given pixel patch is likely to recur within the same image. This multi-scale self-similarity is an intuitive generalisation; for example, repeating textural patterns like wallpaper vary in size as the distance to the capturing camera changes. The SP is computed that characterise multi-scale self-similarity as follows:

A reference texture patch of size $w \times w$ pixels is first selected in a texture image. Then each sliding window of $(w + \beta) \times (w + \beta)$ pixels is resized to $w \times w$ pixels where β denotes the scale values within a given scale range. Let TP represents the target patch corresponding to which the best matched candidate patch (CP) needs to be identified, thus TP^i can be defined as:

$$TP^i = \{TP^1, TP^2, ... TP^m\} \tag{5.1}$$

Where $i = 1$ to m represent the number of reference TP. To find the CP^i for each TP^i, search space S^j_β is generated by resizing the patches for all values of β as:

$$S^j_\beta = \{S^1_\beta, S^2_\beta, ..., S^n_\beta\} \tag{5.2}$$

Where $j = 1$ to n, represent the number of scaled patches in the generated search space. Using MSE as the distortion metric, for each β, the number of best matched CP is identified as:

$$CP^i_\beta = \min_\beta MSE(S^j_\beta, TP^i) \quad where \quad 1 < i < m \; and \; 1 < j < n \tag{5.3}$$

$$Total\, Number\, of\, best\, CPs = sum(CP^i_\beta) \quad where \quad 1 < i < m \tag{5.4}$$

The range of β values for which the total number of best matched patches is higher than threshold value T defines the SP that characterises, multi-scale self-similarity in this image.

5.3 Self-similarity Characterisation based JTDI

Having defined the notion of multi-scale self-similarity in natural images, now the processing blocks of proposed SC-JTDI are discussed. There are two main blocks namely *multi-view encoder* and *multi-view decoder* on either side of the transmission block as shown in Figure 5.1. The scale-based self-similarity characterisation

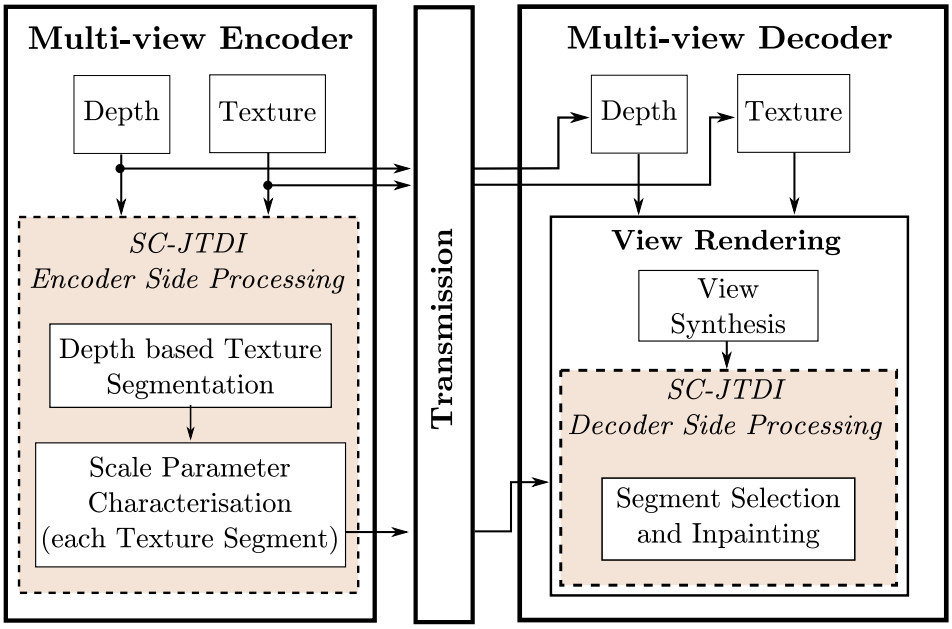


Figure 5.1: Block diagram of SC-JTDI.

is performed at the encoder while the inpainting process is carried out at the decoder. The encoders, in general, are computationally more powerful than decoders (Lukac, 2012). Thus, performing the characterisation at the encoder aims to avoid any additional computation load at individual user-ends.

At encoder an image is segmented into multiple depth segments using available per-pixel depth values. Followed by segmentation, each segment undergoes scale range characterisation for self-similarity analysis as discussed in Section 5.2. The computed SP for all segments are then transmitted as *supplementary information* (SI) to the decoder. At decoder, disocclusion holes are inpainted by performing TM on per-segment basis and searching for similar patches with the designated SP. The segmentation at decoder is intended to identify and employ the BG segment as a dedicated search space for TM. Since the characterisation performed at the encoder decides the parameters that contribute to the disocclusion hole-filling at decoder, it is termed as *encoder-guided strategy*. The following sub-sections describe the

operations at the encoder to firstly characterise self-similarity of camera-captured texture images; and secondly the operations at the decoder to perform encoder-guided inpainting.

5.3.1 Encoder Side Processing

At the encoder, the objective is twofold : i) segment the camera-captured texture image into depth segments; contiguous spatial areas with similar depth values, and ii) define and transmit SP for each depth segment to the decoder for encoder-guided disocclusion inpainting. Figure 5.2 shows a detailed block diagram of encoder side processing and is explained as:

Step (1)**: Depth and Texture Segmentation**

The goal of depth segmentation is to divide a camera-captured texture image into contiguous spatial areas that roughly correspond to physical objects in the 3D scene. The segmentation aims to reduce complexity at the decoder by performing multi-scale TM on per segment basis instead of per image.

This is reasonable, since repeated textural patterns likely recur within the same physical object, contained in a depth segment. However, there may appear more challenging cases where multiple objects occur within same depth segment. Let I_T and I_D denote the texture and depth maps of a camera-captured reference view, respectively. First I_D is divided into s segments by detecting peaks and valleys in a constructed histogram of depth values using default values as in (Silva et al., 2010). Figures, 5.3 (a) and (b) represent depth histograms for *Aloe* and *Cones*

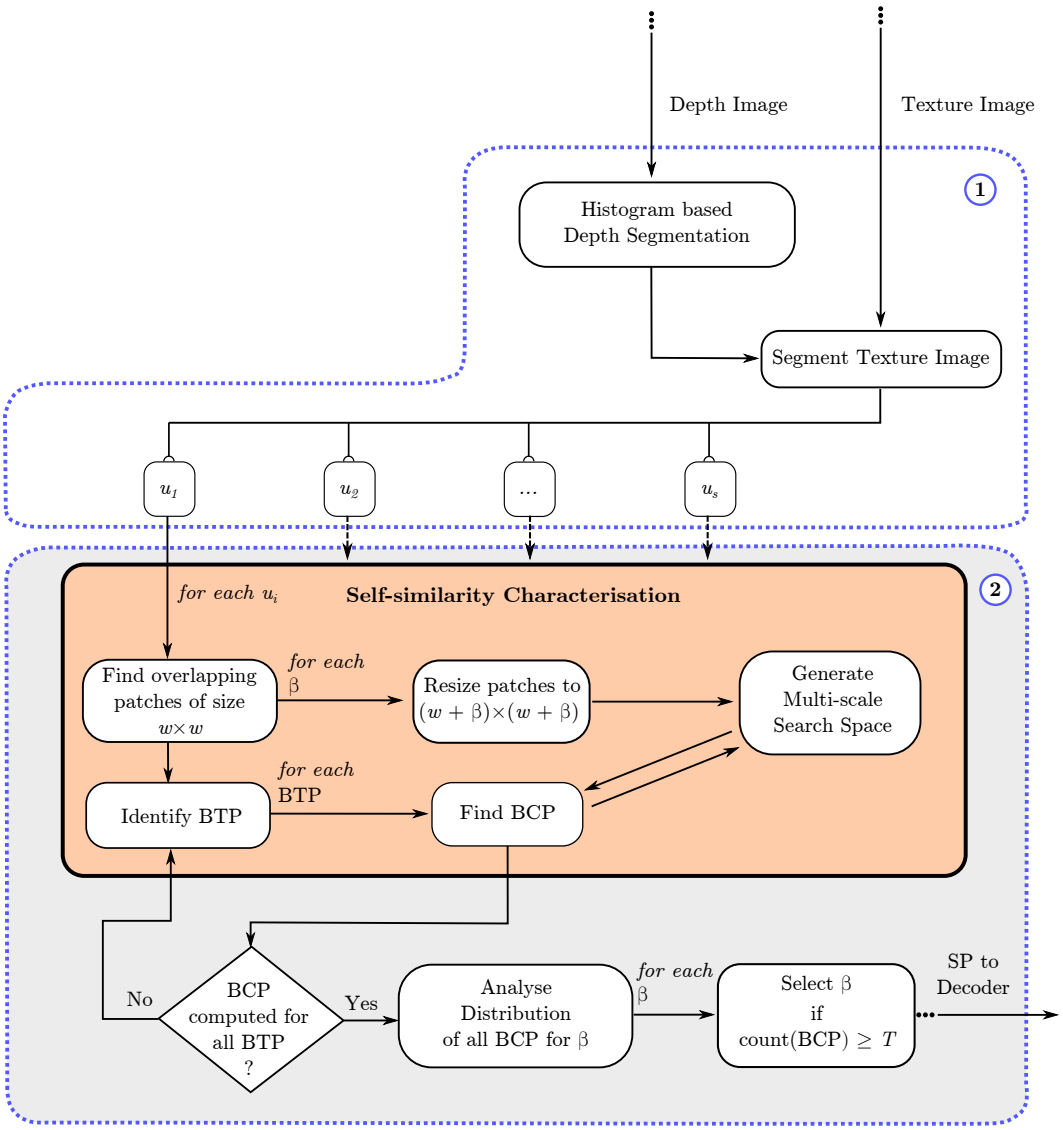


Figure 5.2: SC-JTDI: Encoder side processing with contribution highlighted in step (1) and (2).

datasets respectively. The red line represents the depth cut-off used for segmentation based on local minima. The depth cut-off values $D = \{z_i\}$ correspond to the depth segments and these cut-off values are used to perform the segmentation of corresponding I_T and the segments are given as $U = \{u_i\}$ $where\, i = 1, 2 ..... s$.

The resulting segmentation results for *Aloe* and *Cones* are shown in Figure 5.4 and Figure 5.5, respectively. Figures, 5.4 (a) and (b) show depth segment 1 and segment 2 for *Aloe,* and its corresponding texture segment 1 and segment 2 are

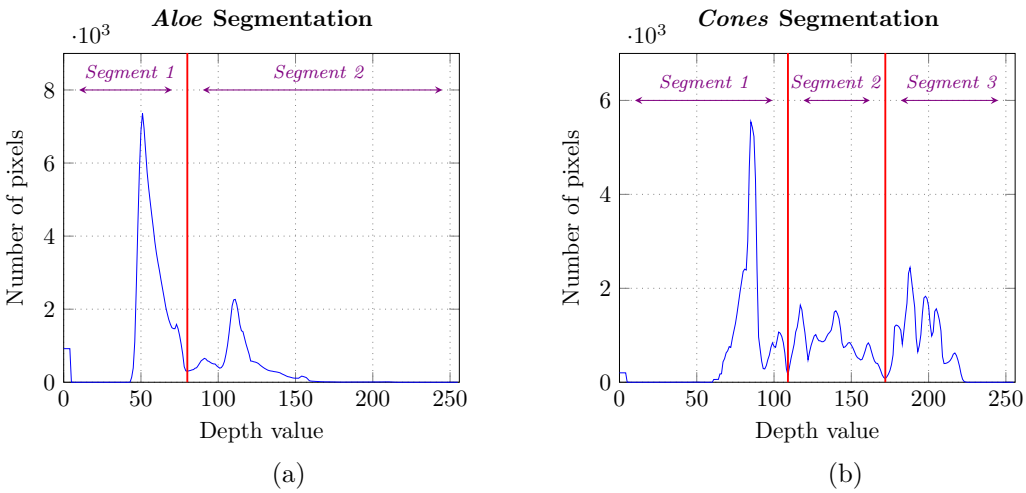


Figure 5.3: Depth-based histogram for (a) *Aloe* and (b) *Cones* dataset.

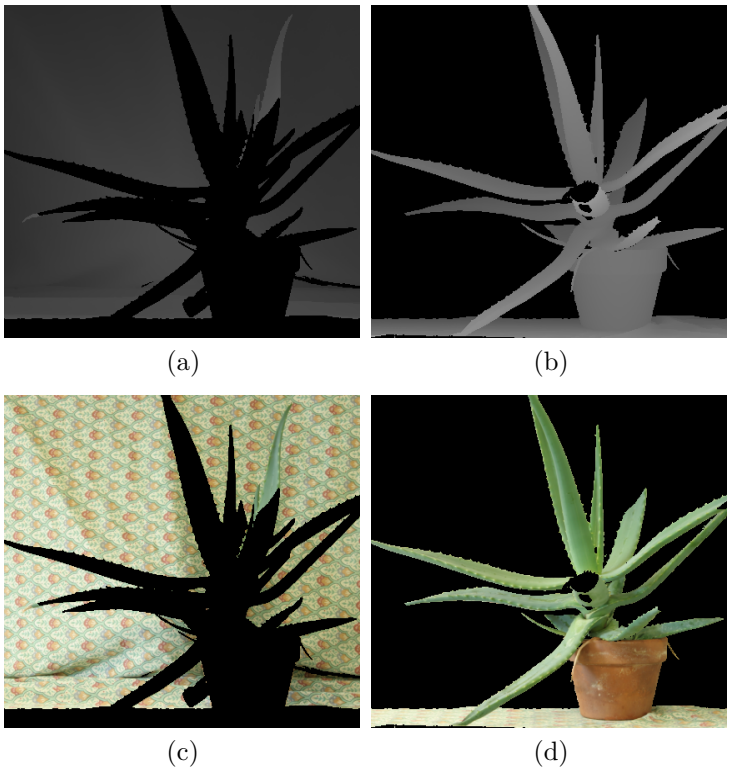


Figure 5.4: Segmentation results for *Aloe* dataset (a) depth segment 1 (b) depth segment 2 (c) texture segment 1 and (d) texture segment 2.

shown in Figures, 5.4 (c) and (d). The BG segment (i.e. segment 1) for *Aloe* is a patterned texture and FG segment (segment 2) represents almost homogeneous region (i.e. plant). However, unlike *Aloe*, the texture segment results for *Cones* in Figures 5.5 (d), (e) and (f) contain different image characteristics in various

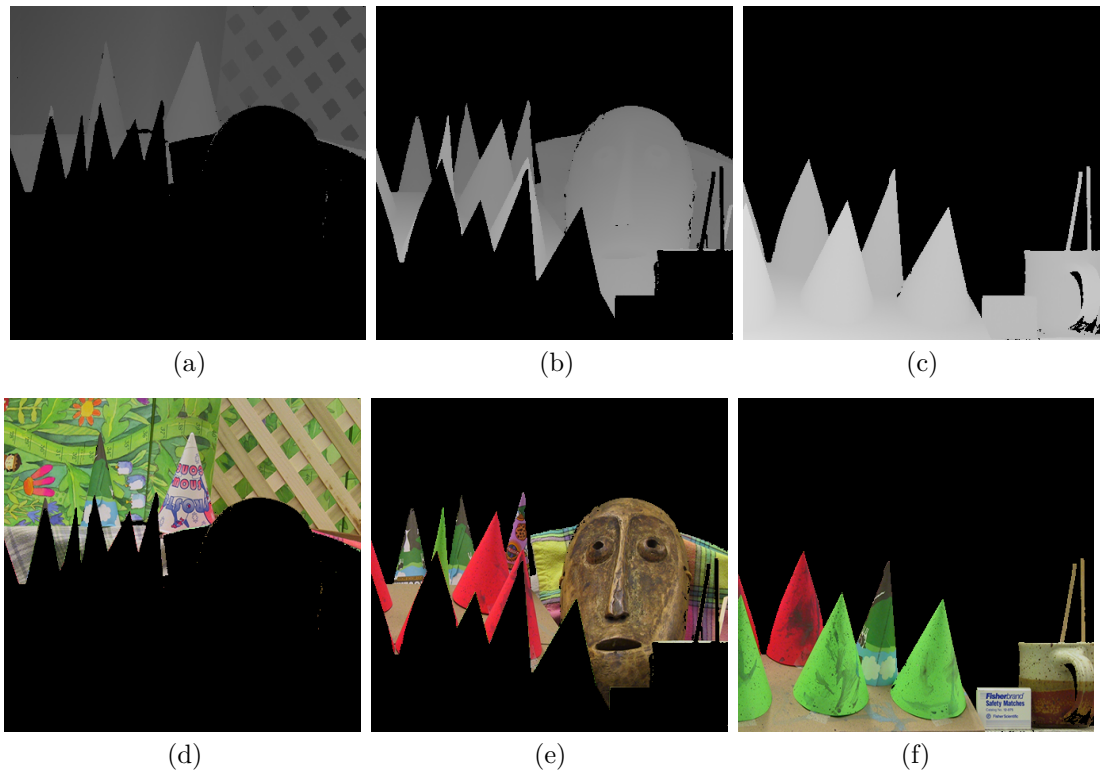


Figure 5.5: *Cones* dataset (a) depth segment 1 (b) depth segment 2 (c) depth segment 3 and (d) texture segment 1 (e) texture segment 2 and (f) texture segment 3.

segments like varying patterned region in segment 1, multiple objects in segment 2 and segment 3. The different characteristics in these datasets provided a motivation for considering them for self-similarity characterisation and evaluating their performance for disocclusion hole-filling.

Step (2): Scale Parameter Characterisation

In this step, the multi-scale self-similarity for each computed texture segment is characterised as discussed in Section 5.2. In these experiments, the scale value β is considered within range $[-3; 3]$ i.e.$[-3, -2, -1, 0, 1, 2, 3]$. The patches at various β are resized and compared against specific *boundary texture patches* (BTP) to compute the MSE. Figure 5.6 represents the BTP (shown in *red* boxes) which are

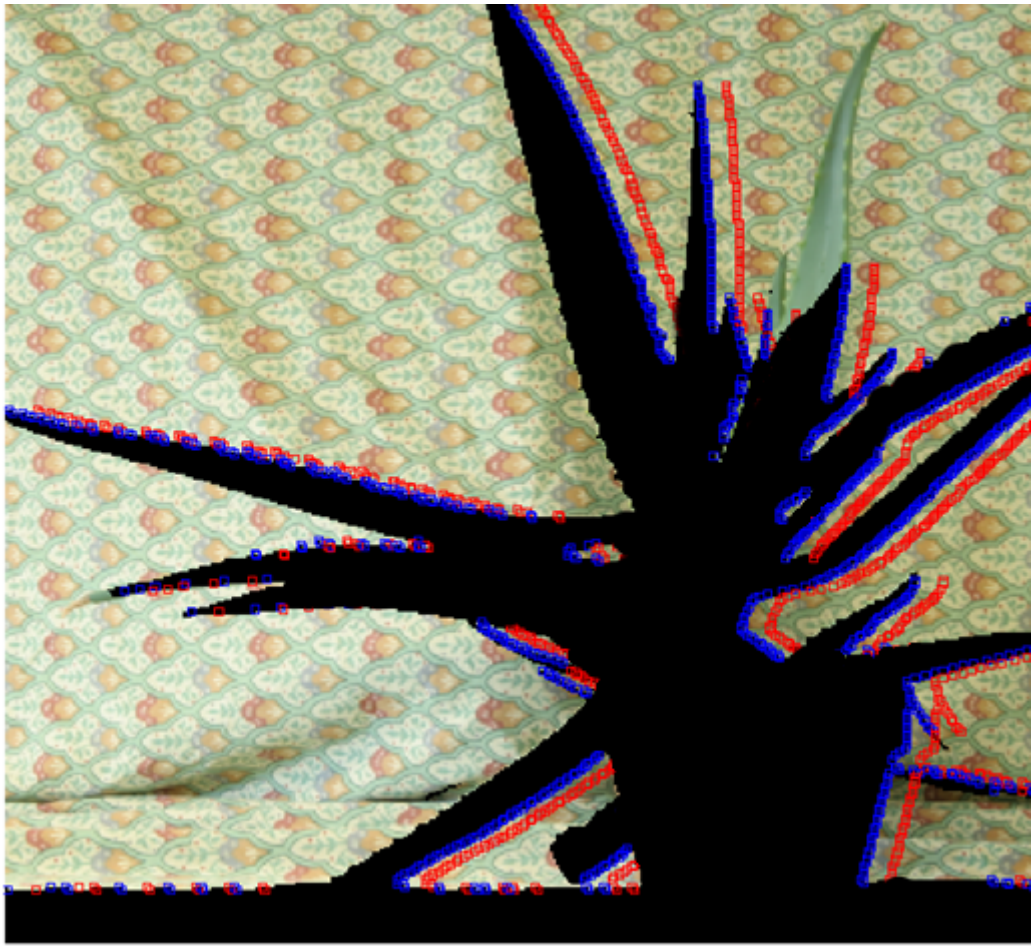


Figure 5.6: *Aloe* with reference texture patches (in *red*) near boundary (in *blue*).

considered near the hole boundary regions using a sliding window mask. These BTPs are used for error computation to find the *best candidate patches* (BCP). Such a choice arises from the fact that disocclusion holes tend to appear near object boundaries (Tian et al., 2009).

For designating the best SP to a given segment, it is thus logical to match the scaled patches against the BTP instead of matching with patches from whole image that tends to be more time-consuming. Once the BCP are determined using (5.3) and (5.4), the bar graph is plotted to analyse those values of β which produced most candidate matches. This is achieved by comparing these β values against T and selecting all the β values above T as the SP for given segment. Figure

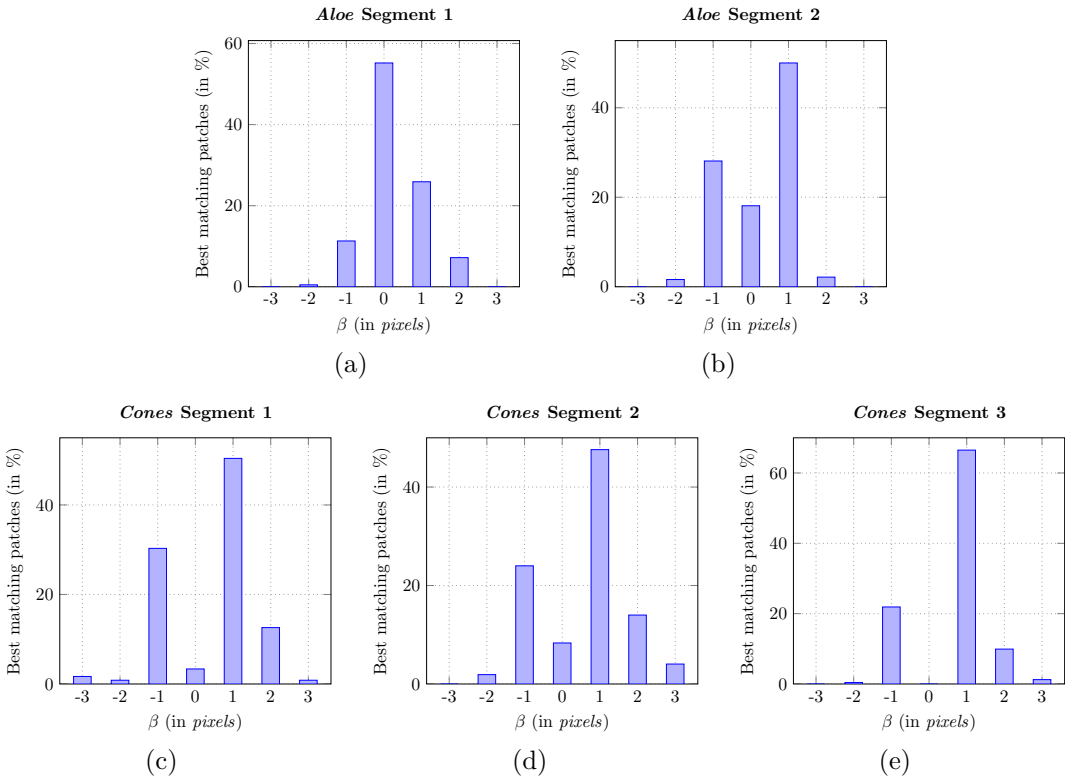


Figure 5.7: Bar graphs for *Aloe* Segment 1-2 in (a) & (b) and *Cones* Segment 1-3 in (c), (d) and (e) respectively.

5.7 illustrates the number of best matches for various β values within the range $[-3; 3]$ for *Aloe* and *Cones* images. For a given segment, if the percentage of the best matching patches exceeds a threshold T, the corresponding β value is included as SP where the value of T is empirically chosen as 20% based on the bar graphs plotted for various datasets. For example, *Aloe* segment 1, $\beta = 1$ and 0 yield more than 20% of best matches and thus are chosen as SP for segment 1 whereas for segment 2, $\beta = 1$ and -1 is selected as its corresponding SP. This implies a high possibility of finding a superior match at SP = [0; 1] and [1; -1] while inpainting holes in segment 1 and segment 2, respectively. It is also observed that although there have been few good matches at $\beta = 2$ and -2, but they are excluded from SP since their best patch percentage is below the set threshold level.

However, there are scarcely any matches at $\beta = 3$ and -3, which state the unavailability of sufficient self-similar patches at these scale values. Selecting the scale values below the threshold will cause additional computation cost without significant advantage in inpainting process. Similarly, for *Cones*, it is observed that SP = [1; -1] is dominant throughout all 3 segments. The segmentation results for more datasets are included in Appendix C (shown in Figures C.1 - C8).

The chosen SP for each segment and depth cut-off values are transmitted as SI to the decoder along with the reference views for encoder-guided disocclusion inpainting. The SI transmission accounts for only a small signalling overhead compared to the size of the reference texture and depth maps. This analysis is not provided as it is considered beyond the scope of this work.

5.3.2 Decoder Side Processing

The decoder receives I_T, I_D , SP per segment and their cut-off values D. In previous chapter, JTDI inpainted texture and depth hole pixels alternately: first using available depth information to fill in textural pixel holes, and then use inpainted textural information to fill corresponding depth pixel holes. However, JTDI employs full-image for TM, to search best CP for each TP as shown as step (**2**) in Figure 4.1. It was observed that such search process may select the CP from the FG which results in error propagation and also since inpainting of holes is an iterative process, the exhaustive search process becomes highly time-consuming.

The SC-JTDI adopts the joint texture and depth inpainting technique but intends to minimise the errors due to CP selection from FG and reduces the in-

painting time by employing multi-scale TM within suitable depth segment instead of full image. The SP provides information on the suitable scaling values to resize the patches for each segment and generate a superior search space for TM.

The virtual texture view (V_T) and virtual depth map (V_D) are synthesised from I_T and I_D via DIBR. Before filling disocclusion holes, bothV_D and V_T are segmented with the same cut-off values D as discussed in Section 5.3.1. The following sub-section explains the steps involved in decoder side processing as shown in Figure 5.8 for SC-JTDI.

Step (1)**: Compute Priority**

The depth-based priority order to select the TP is adopted from JTDI (see step (1) in Figure 4.1). The selected TP is used to identify the segment which is used to generate the multi-scale candidate search space for finding the CPs in step (2).

Step (2)**: Segment Selection and Template Matching**

The target depth patch (Z_p) corresponding to texture TP is selected and its known depth values are used to compute the depth mean ($\overline{Z}_p$). Since disocclusion holes are missing pixels from BG region, $\overline{Z}_p$ facilitates the selection of appropriate BG depth segment(s) U_b as follows:

$$U_b = \{u_i \in U \mid \overline{Z}_p \leq z_i,\, z_i \in D\} \quad where \quad i = 1,\, 2,\, \ldots,\, s. \tag{5.5}$$

The SP corresponding to the selected depth segment U_b helps in generating

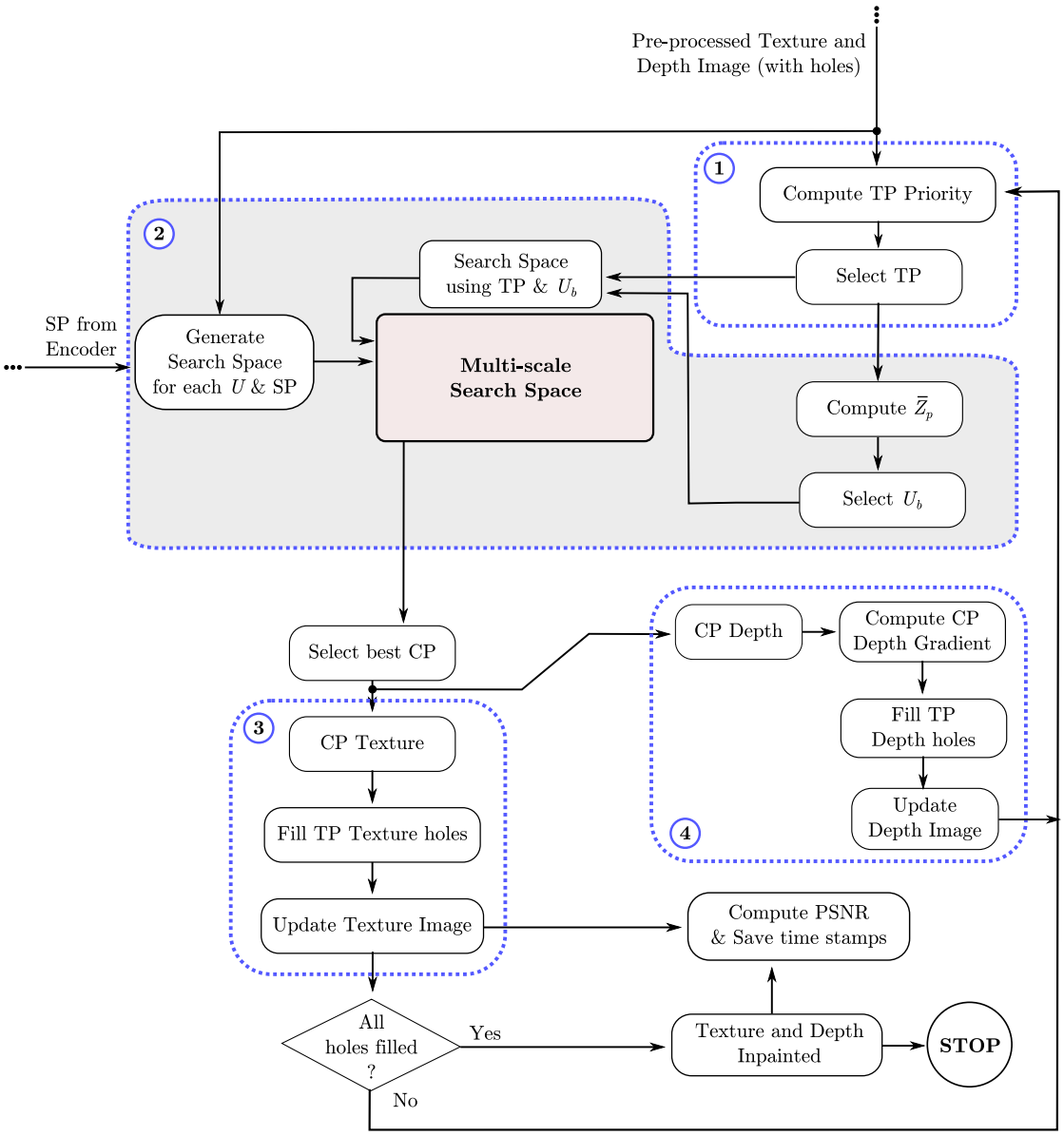


Figure 5.8: SC-JTDI: Decoder side processing with contribution highlighted in step (2).

multi-scale candidate search space X by resizing the patches for given SP values in the range such that $X = \{x_1, x_2, \ldots, x_h\}$ where h represents number of patches in X. This search space is used for finding best candidate patch CP as follows:

$$CP = min\, MSE\,(x_j; TP) \quad where \quad j = 1, 2, \ldots, h. \tag{5.6}$$

Step ③ **and** ④**: Texture and Depth Inpainting**

After the selection of CP, the known pixels of the CP are then copied into corresponding unknown (holes) pixels of TP and the depth holes are inpainted as described in JTDI (see step ③ and ④ in Figure 4.1). This process repeats until all the disocclusion holes are filled.

Thus, instead of employing exhaustive search, the search space is confined to a segment but is enriched with addition of more reliable patches in multi-scale search space generated using SP. The next section discusses the experimental set-up and results for SC-JTDI.

5.4 Experimental Set-up and Results

In order to evaluate the performance of SC-JTDI, *eight* Middlebury image datasets are used. This section discusses in-depth *two* of these datasets, namely *Aloe* and *Cones,* which are chosen for their distinctive features as already discussed in Section 5.3. The results for remaining datasets are included in Appendix E. For each dataset, reference view #1 is used to generate the view #3.

For quantitative and qualitative performance evaluation, the generated view #3 is inpainted using SC-JTDI with $w = 9$ and compared against the JTDI results. The comparator JTDI employs single-scale exhaustive TM to find CP for filling disocclusion holes. For numerical analysis, the original view #3 of image datasets is used as the ground truth for the PSNR calculations, and the PSNR is computed for both the whole image and inpainted region.

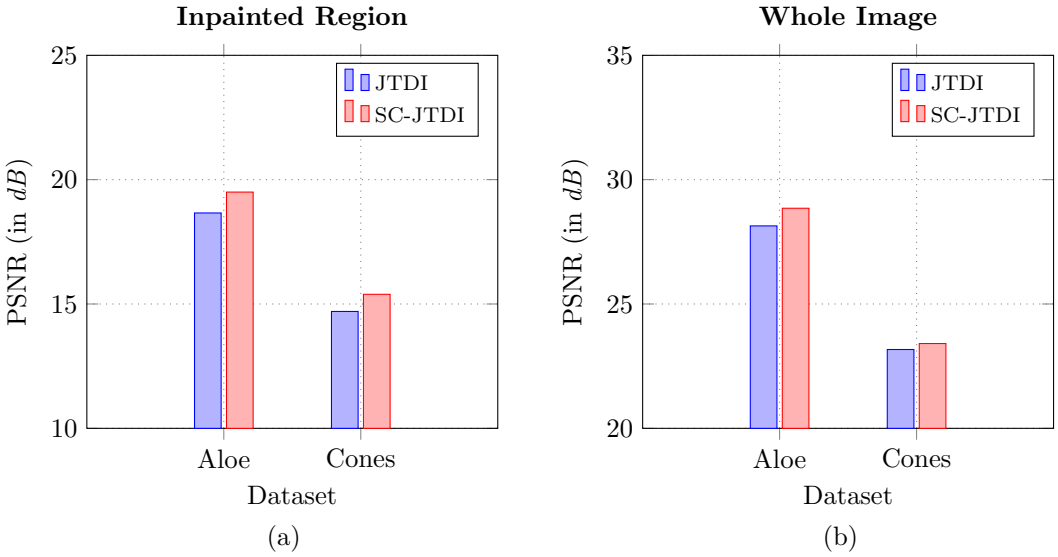


Figure 5.9: PSNR comparison for *Aloe* and *Cones* datasets in two scenarios namely, (a) Inpainted Region and (b) Whole image for JTDI and SC-JTDI respectively.

5.4.1 Quantitative Result Analysis

Figure 5.9 shows the PSNR results for two inpainting methods, which reveal that SC-JTDI performs consistently better than JTDI. For *Aloe* and *Cones*, the PSNR increased by 9.41% and 3.52% for inpainted region in comparison to JTDI. This shows that characterisation parameters provided as SP results in generating superior space which provides better matching patches with lower MSE. The increase in PSNR supports the fact that during the disocclusion inpainting process, there are cases where self-similar patches are available at scales other than $\beta = 0$ (same scale) and provide better matched patches which leads to high PSNR.

5.4.2 Qualitative Result Analysis

From a perceptual quality perspective, the qualitative comparison of proposed SC-JTDI is performed against JTDI. For comparison, the disocclusion holes, ground

truth, JTDI and SC-JTDI results are shown for *Aloe* and *Cones* datasets in Figure 5.10 and Figure 5.11 respectively. The areas marked red in Figure 5.10 (a) and Figure 5.11 (a) highlight part of the problem regions which contain larger holes and are challenging to inpaint. The zoomed-in areas for *Aloe* and *Cones* are shown in Figure 5.10 (b) and Figure 5.11 (b). On comparing the results of JTDI and SC-JTDI against the ground truth, it is observed that SC-JTDI results in Figure 5.10 (e) and Figure 5.11 (e) are considerably better as compared to JTDI in Figure 5.10 (d) and Figure 5.11 (d). The zoomed-in region for *Aloe* refers to the same region considered in Chapter 4 for analysis. It is observed that patterned BG region near the leaf edges of inpainted *Aloe* region is well-recovered using SC-JTDI (see Figure 5.10(e)) but inherits artefacts due to FG region filling in JTDI (see Figure 5.10(d)). The BG in *Aloe* is a repetitive pattern, thus enhancing the search space by introducing scaled patches during TM resulted in better matches and provide improved inpainting. The segment based search however restrained the search space to the BG region and avoid the CP selection from the FG.

Similar trend can be seen in *Cones* datasets shown in Figure 5.11(e) where the details at the object boundaries are well-inpainted and preserved better with fewer artefacts in comparison to JTDI. From Figure 5.11 (d) and (e), it is observed that although there is no pattern like *Aloe* but SC-JTDI successfully filled in the missing information similar to the original image (Figure 5.11 (c)). This shows that multi-scale search space helped in retrieving the non-patterned textures as well and proves the robustness of the proposed technique in different scenarios. The artefacts in JTDI result from full-image exhaustive TM and its inability to find good match due to scarcity of potential patches in search space. Once the

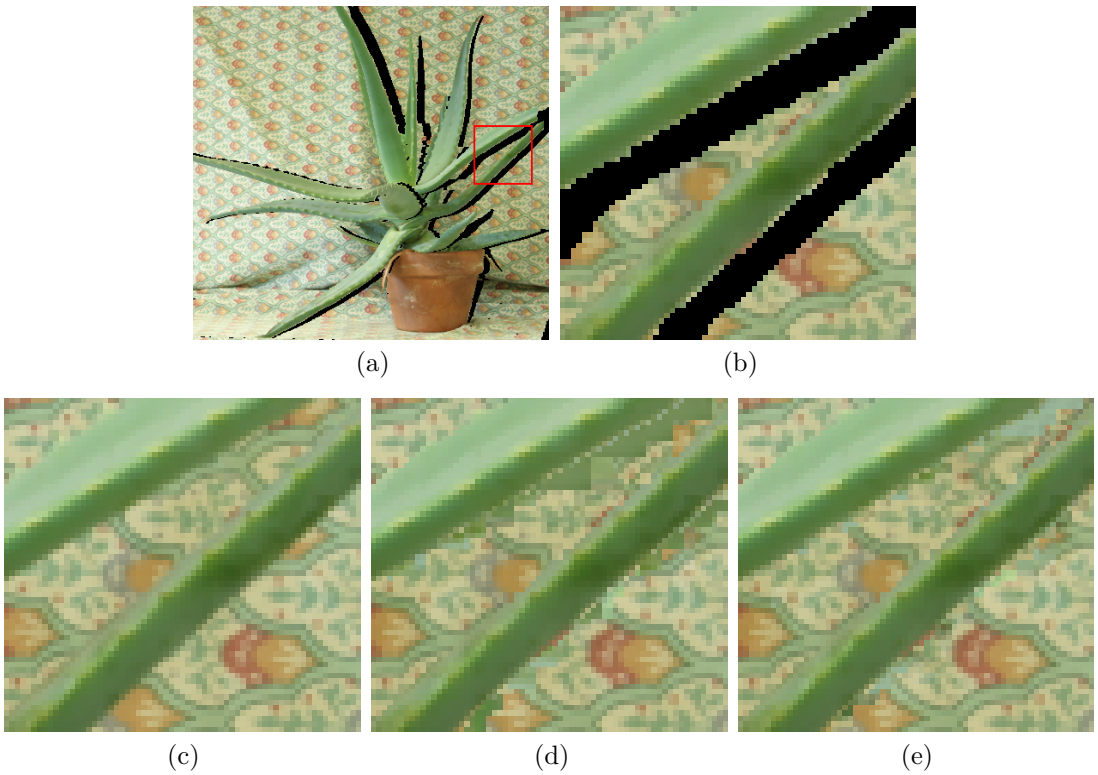
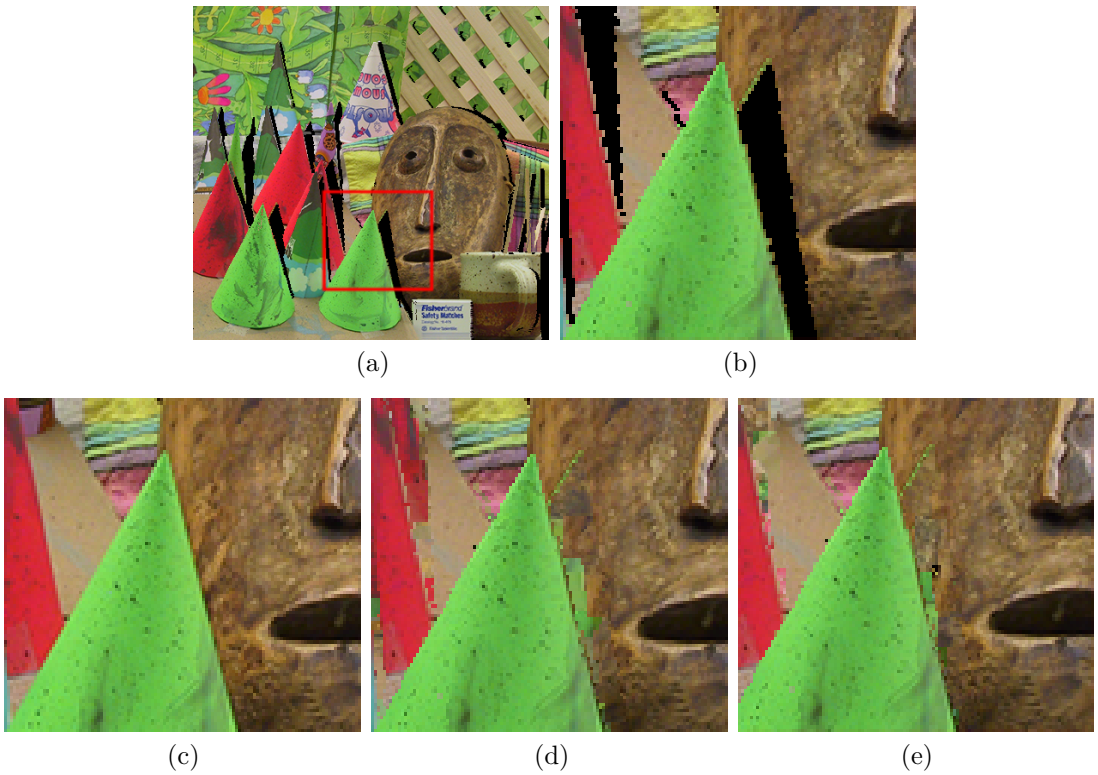


Figure 5.10: *Aloe* (a) Image with holes (b) Holes sub-region, (c) Ground truth, and (d) and (e) represent inpainting results by JTDI and SC-JTDI respectively.

patch is wrongly filled, it led to increased error propagation further in the filling process. Unlike JTDI, SC-JTDI search for the CP only in a selected segment as mentioned in (5.5) and (5.6) and rejects most of unwanted patches. Similar trend is observed throughout the experimentation for other datasets, as presented in Appendix E.

Overall, it is observed that SC-JTDI achieved improved visual quality with fewer inconsistencies and better preserves the FG object boundaries in comparison to JTDI. Multi-scale TM reduces the artefacts and fills the disocclusion holes providing enhanced perceptual quality.



(a) (b)

(c) (d) (e)

Figure 5.11: *Cones* (a) Image with holes (b) Holes sub-region, (c) Ground truth, and (d) and (e) represent inpainting results by JTDI and SC-JTDI respectively.

5.4.3 Inpainting Time Analysis

This section discusses the inpainting involved in SC-JTDI in comparison to the previous JTDI. At this stage it is important to clarify that the computation performed for segmentation and scale characterisation at encoder is entirely offline and the inpainting time discussed is purely on the basis of time involved in inpainting the texture and depth holes at the decoder.

Figure 5.12 represents time performance plots of both *Aloe* and *Cones* datasets for SC-JTDI and JTDI inpainting. In terms of the inpainting time, SC-JTDI shows improved time as compared to JTDI. The computation cost due to multi-scale TM is compensated by segment based search during inpainting and offline search space generation. Though searching for patches of different scales entails a larger search

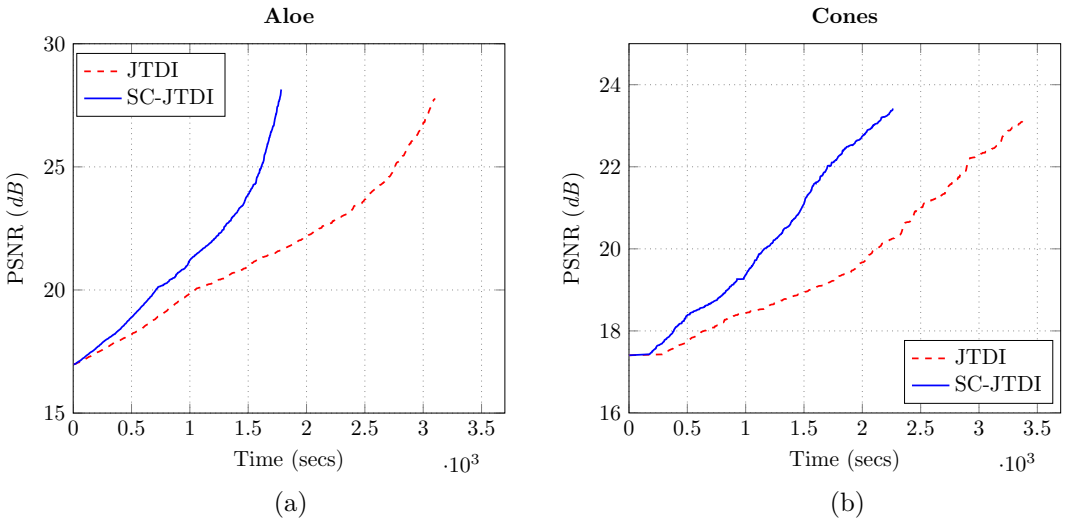


Figure 5.12: Time performance comparison plots for (a) *Aloe* and (b) *Cones*.

space, the resulting search complexity is contained by performing TM only within designated depth segments i.e. subset of the image with similar depth values.

Thus, the overall computation time is considerably decreased as compared to exhaustive TM used in JTDI. The time saving for *Aloe* dataset is 46.5%, on similar grounds the time involved in inpainting *Cones* is reduced by almost 24%. The inpainting time can exceed in case more scale values are added to the SP for increased search space. Using this approach, considering the amount of time saving while employing multiple scale values for inpainting and an increased PSNR justifies the impact and reliability of SC-JTDI.

5.5 Summary

In this chapter, a fast SC-JTDI is proposed that exploits multi-scale self-similarity in combination with the encoder-guided strategy for inpainting disocclusion holes. The segment based self-similarity characterisation is performed at encoder to save

additional computation at the individual decoder end. At decoder, inpainting is performed within suitable depth segments but across multiple scales as specified by the transmitted self-similarity parameters. Experimental results show that proposed technique outperforms JTDI by providing improved visual and numerical performance. The extension of search space through additional scaled patches is compensated by constraining the search space to selected texture segments which led to superior inpainting as well as increased inpainting speed. The results demonstrate the significant contribution of scale-based self-similarity characterisation and its effectiveness in disocclusion inpainting.

Although the performance of SC-JTDI is appreciable both numerically and visually, the analysis shows that the choice of scale parameters influence the overall inpainting performance. The self-similarity characterisation is based upon the scale range which is selected empirically as $[-3; 3]$. It is evident from the detailed discussion in Section 5.4 that the choice of SP is highly dependent on image characteristics and have a tendency to vary for different images. Thus further investigation is required to characterise the self-similarity automatically such that it does not require an empirical scale range and is capable of appropriately choosing the best scales for a given image. Apart from scale-based self-similarity, natural images tend to possess other similarities such as rotation. This provides a motivation to explore advanced characterisation methods to incorporate additional self-similarity for more robust inpainting.

Chapter 6

Advanced Self-similarity Characterisation based JTDI

6.1 Introduction

The previous chapter discussed a new encoder-guided strategy that exploited multi-scale self-similarity to enhance the candidate search space for improving the overall inpainting performance. Though searching for non-local patches of different scales entailed a larger search space, the resulting search complexity was contained by performing TM within designated depth layers. The scale-based, self-similarity characterisation has proven valuable as discussed in the previous chapter; however, the presence of spatial similarities may vary depending on the inherent image characteristics. This provided the motivation to investigate whether it is possible to devise a technique capable of detecting additional self-similarities present in an image and then to employ these while inpainting the virtual views.

It is observed that another way of characterising the self-similarity contained within an image is rotation i.e. the occurrence of similar patches in an image at different angles. This chapter extends the concept of Self-similarity by utilising rotation-based self-similarity analysis along with scale to determine the Scale and Rotation (SR) parameters which can be utilised for inpainting disocclusion holes.

Chapter 5 introduced self-similarity characterisation with an empirical assumption concerning the scale range. In contrast, this chapter introduces an Advanced Self-similarity characterisation (ASC) to automatically determine scale and rotation parameters combinedly for inpainting disocclusion holes. ASC exploits scale and rotation invariant properties of Log-Polar Transform (LPT) together with scale and rotation angle detection technique of Fourier Mellin Transform (FMT) to achieve self-similarity characterisation.

6.2 Advanced Self-similarity Characterisation

In certain circumstances, two self-similar patches can lead to high MSE, having undergone a small geometric transformation in terms of rotation and scale and can prevent its selection as a potential CP during TM. This chapter aims to include the scaled and rotated self-similar patches as possible candidates during TM for inpainting disocclusion holes. To achieve this, ASC is performed which is designed as a two-step approach: 1) Detects self-similar patches using LPT, which are scale and rotation invariant and 2) compute scale and rotation values among the correlated patches by applying FMT.

LPT is a renowned approach for its rotation and scale invariant properties (Araujo and Dias, 1996; Matungka, 2009; Wong et al., 2008). It uses Log-Polar coordinates representation instead of Cartesian coordinate which represents the rotation and scale in the Cartesian coordinates as shifting in the angular and log-radius directions in the log-polar coordinate, respectively. A point $(x,\, y) \in \mathbb{R}^2$ in the Cartesian coordinates is mapped to log-polar coordinates ρ, θ as:

$$\rho = log\sqrt{x^2 + y^2} \tag{6.1}$$

and

$$\theta = tan^{-1}\frac{y}{x} \tag{6.2}$$

Further details about LPT are provided in appendix F.1. LPT have been widely used in the literature for pattern recognition (Traver and Pla, 2003), face detection and tracking (Jurie, 1999), texture classification (Mahersia and Hamrouni, 2008), image registration (Reddy and Chatterji, 1996; Zokai and Wolberg, 2005), forgery detection in digital images (Bravo-Solorio and Nandi, 2011; Myna et al., 2007) etc. This provides the motivation to utilise the scale and rotation invariant property of LPT for self-similarity characterisation and apply it for inpainting disocclusion holes. The details of self-similarity detection using LPT and FMT are provided below:

6.2.1 Self-similarity detection using LPT

In the literature, the LPT based method has been used to detect duplicate regions affected by reflection, rotation and scaling in image forensics (Bravo-Solorio and Nandi, 2011) and image watermarking (Lin et al., 2001). To achieve the detection

of similar patches, overlapping pixel patches of an image are converted into rotation and scale invariant log polar maps (LPM). The sum of LPM along the log-radius axis results in the 1D descriptor which affords an efficient search for self-similar patches. These computed 1D descriptors are invariant to both scaling and rotation.

A 1D descriptor $\overrightarrow{g}_i$ corresponding to a grey-scale patch of pixels $A_i(x, y)$ is given by:

$$\overrightarrow{g_i}(\rho) = \sum_{\theta} A_i(\rho, \theta) \tag{6.3}$$

and the corresponding descriptor of rotated and scaled version, $A_i^{'}$ is represented as:

$$\overrightarrow{g}^{'}_i(\rho) = \sum_{\theta} A_i^{'}(\rho, \theta) \tag{6.4}$$

. In reference to the well-known translation properties of the Fourier Transform (FT) (Bracewell, 1999), the Fourier magnitude of both descriptors should be closely correlated as:

$$c(\overrightarrow{G}_i, \overrightarrow{G}^{'}_i) = \frac{\overrightarrow{G}_i^T . \overrightarrow{G}^{'}_i}{\sqrt{(\overrightarrow{G}_i^T \overrightarrow{G}_i)(\overrightarrow{G}_i^{'T} \overrightarrow{G}^{'}_i)}} \tag{6.5}$$

where c is the correlation coefficient, $\overrightarrow{G}_i$ and $\overrightarrow{G}^{'}_i$ are the Fourier magnitudes of $\overrightarrow{g}_i$ and $\overrightarrow{g}^{'}_i$, while superscript T represents the transpose, respectively. If the correlation coefficient of two descriptors is close to 1, that implies they are closely correlated and similar. To compute the rotation angle and scale factor among these correlated patches, the FMT (Chen et al., 1994; Raman and Desai, 1995; Sarvaiya et al., 2009) is used.

6.2.2 Fourier Mellin Transform

FMT is a widely used mathematical tool for image recognition as its resulting spectrum is invariant to rotation, translation and scale (Panigrahi, 2014; Singh et al., 2005). Since, FT is translation invariant, its conversion to log-polar coordinates converts the scale and rotation differences to vertical and horizontal offsets which can then be measured. FMT combines the phase correlation with LPT to quantify the scaling, rotation and translation parameters among two correlated outputs (Reddy and Chatterji, 1996).

Firstly, the LPT is applied to the magnitude spectrum of input images, this is because the log-polar transformation manifests rotation and scale as translation. However, the magnitude spectrum of the image and translated image are identical and only their phase spectrum is different, thus the phase correlation is performed in the log-polar space to recover the rotation and scale (Wolberg and Zokai, 2000). Appendix F.2 provides a worked example to further illustrate scale and rotation detection using FMT.

6.3 Advanced Self-similarity Characterisation based JTDI

The block diagram of ASC-JTDI framework is shown in Figure 6.1. The work adopts the encoder-guided strategy discussed in Chapter 5, performing ASC to identify the recurring scales and/or rotation angles per segment at encoder and employing them to generate a superior search space for inpainting at decoder. The

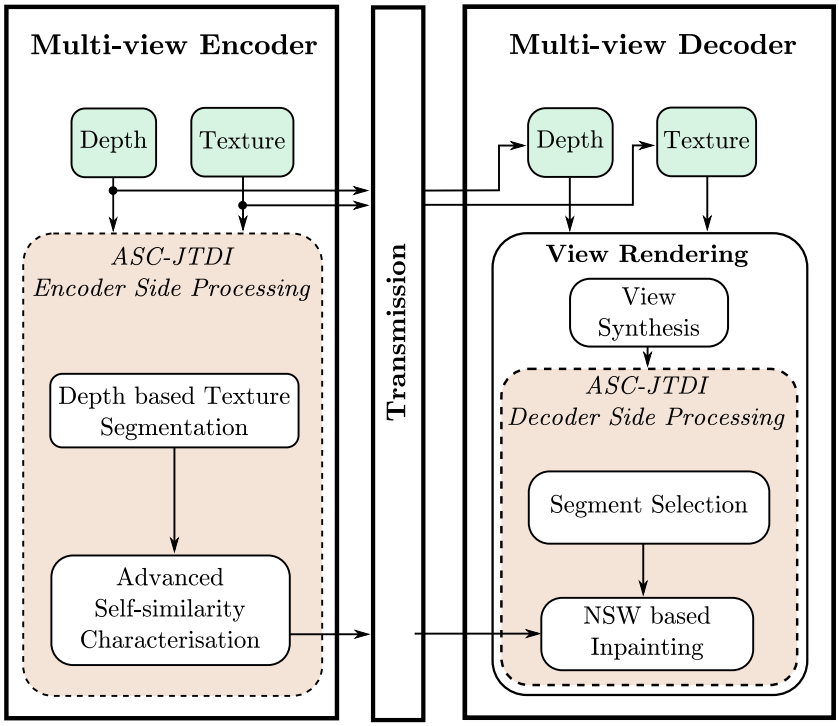


Figure 6.1: Block diagram of ASC-JTDI.

self-similarity characterisation is *advanced* since it automatically determines the most dominant SR parameters contained in individual segments of an image.

The detected SR parameters are then transmitted as SI along with the depth cut-offs and reference views. At the decoder, computed SR parameters are utilised to generate a new Neighbourhood Search Window (NSW) oriented segment search space for TM. The steps at encoder and decoder side processing of proposed ASC-JTDI are illustrated in Figure 6.1 and described in Sections, 6.3.1 and 6.3.2:

6.3.1 Encoder Side Processing

At the encoder, the main aim is to determine SR parameters per segment which can then be transmitted to the decoder to assist the TM for inpainting. This section elaborates upon the various steps involved in encoder side processing for

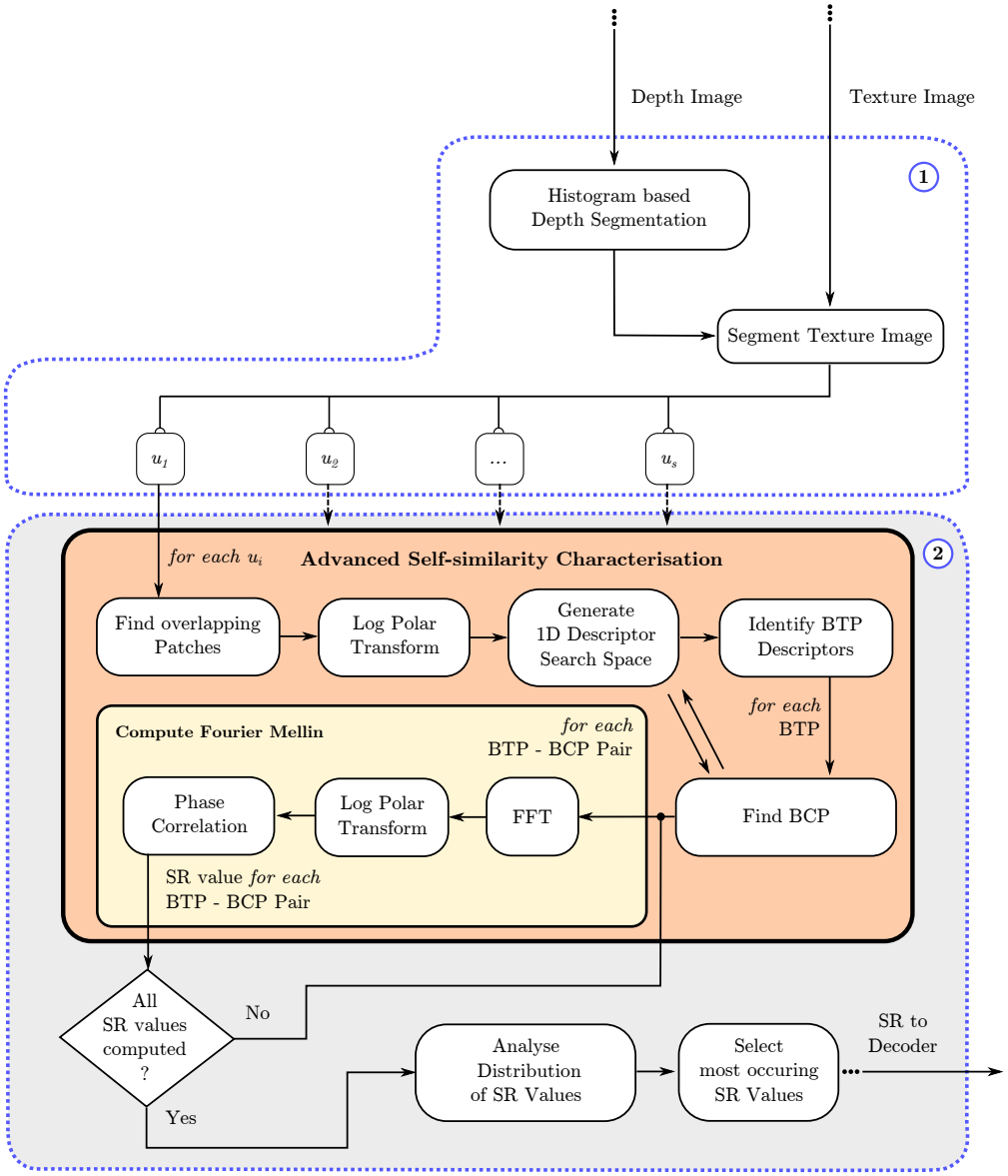


Figure 6.2: ASC-JTDI: Encoder side processing with contribution highlighted in step (2).

ASC-JTDI as shown in Figure 6.2.

Step (1): Depth and Texture Segmentation

Firstly, the histogram based depth segmentation is performed followed by the texture segmentation as discussed in Section 5.3.1 step (1). The segmentation results are similar to Figure 5.2 since the same methodology is applied for segmenta-

tion. Each texture segment undergoes ASC in the next step for SR parameters computation.

Step ② **: Advanced Self-Similarity Characterisation**

The goal of this step is twofold: 1) to detect most similar patch combinations in the segment even if they are scaled and rotated; 2) to quantify the scale factor and rotation angle between the patches for SR parameters computation.

1. **Scale and Rotation Invariant Patch Detection:** To detect the most similar scale and rotation invariant patches, each segment is first divided into overlapping pixel patches. These patches are converted to log polar maps which are used to compute 1D descriptors as discussed in Section 6.2.1. However, the patches with hole pixels are discarded during the mapping. These computed descriptors are rotation and scale invariant. To find the best correlated patches, the descriptors that correspond to the boundary patches are considered as reference patches since these are the significant regions where disocclusion holes tend to occur during view synthesis. The correlation coefficient is computed between the descriptors corresponding to BTP and BCP as in (6.5). The one whose output is closest to 1 is selected as the best correlated patch pair.

   To find the best correlated patches, the following two conditions are imposed to avoid the false matches (Bravo-Solorio and Nandi, 2011):

   (a) Discarding the patches which overlaps with reference patch; such that the minimum distance is $d_{ij} > \tau_d$ where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

and τ_d is equal to the diameter of the patch.

(b) Minimising the patches having low-entropy luminance i.e. patches with uniform or homogeneous information. To achieve this, a luminance feature vector is computed for each patch as:

$$f_i = -\sum_k p_k log_2 p_k \tag{6.6}$$

Where p_k is probability of each luminance value within a patch and the luminance of each colour pixel is computed as (Stone, 2016):

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{6.7}$$

where R, G and B represents *red, green* and *blue* channels in the image. Find the best correlated patches such that $|f_i - f_j| \leq \tau_e$, where τ_e is pre-defined threshold. Imposing these two conditions τ_d and τ_e minimise the occurrence of false matches and results in identifying the true matches which correspond to best correlated patches even when scaled or rotated.

2. **Compute FMT:** The best correlated patch pairs resulting from previous step are used to determine the dominant scale and rotation combinations, in a given segment. The computation of scales and rotation angles is performed using the method discussed in Section 6.2.2. This involves computing the Fast Fourier transform (FFT) for each correlated patch pair detected in previous step and then applying LPT. Employing phase correlation to the log polar transform outputs provides the scale and rotation values between the two correlated patches (Wilmer, 2003). The SR pair for all the correlated

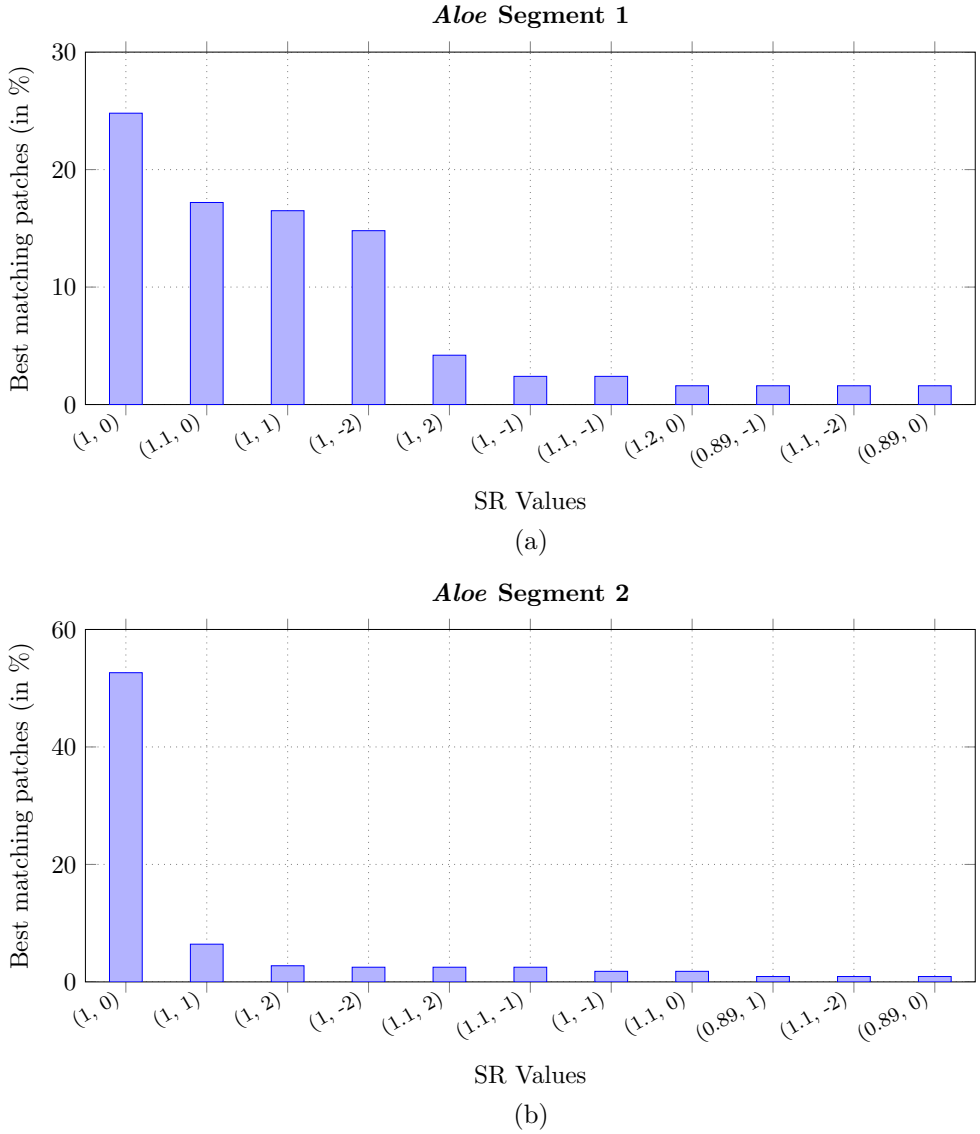


Figure 6.3: Bar graph representing SR (scale, rotation) parameters for *Aloe* (a) Segment 1 and (b) Segment 2. Rotation is denoted in *degrees*.

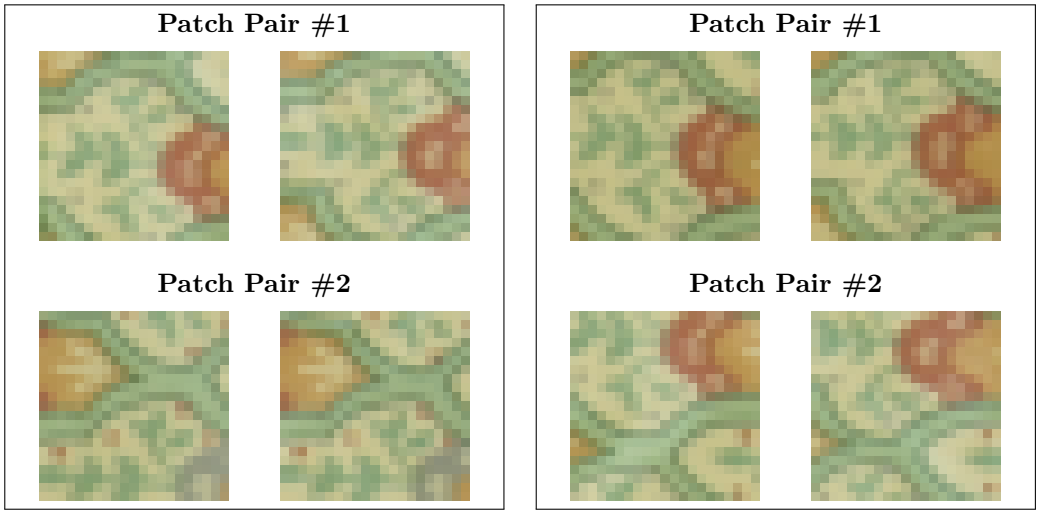
patches is computed to determine SR parameters for each segment of texture image.

3. **SR parameters Analysis:** To analyse the percentage count of each SR pair to determine the most occurring SR pairs, a bar graph is plotted. The SR pairs with the percentage count above a set threshold are selected as the SR parameters for the given segment. These parameters are intended to be employed for generating a search space by resizing the patches for TM

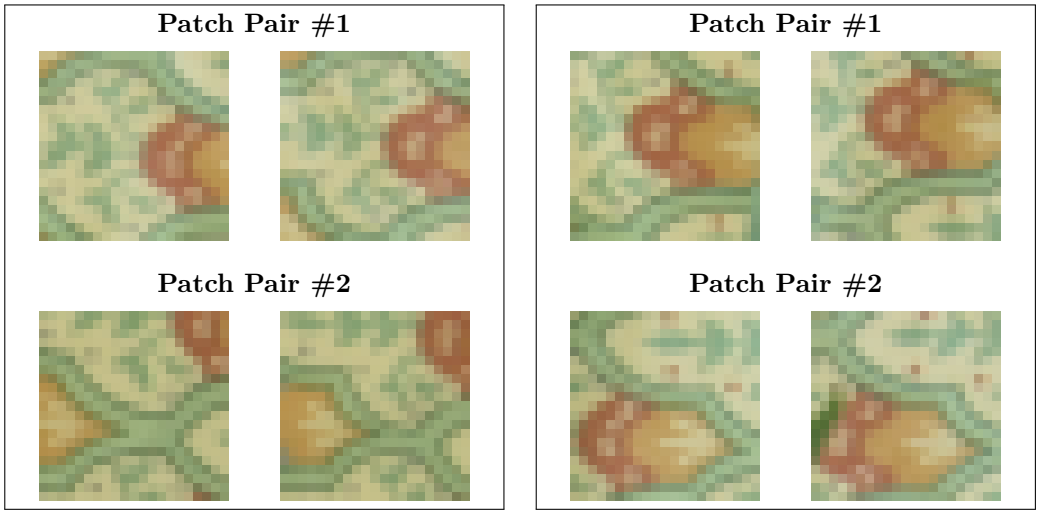
during the inpainting of holes. The bar plots are shown for *Aloe* in Figure 6.3, where the SR pairs are arranged in descending order of the percentage count of best matches on the x-axis. The SR pairs corresponding to fewer patch matches are intentionally removed from the plot since they are insignificant and imply false matches during the SR parameter selection. The threshold value is empirically chosen as 10 % of the best matching patches, which selects a SR pair as SR parameter for a given segment. It is observed that lowering this value increases the inpainting time without a substantial gain in the inpainting performance. The SR pairs for false matches are discarded since they fall below the chosen threshold due their low occurrences. The SR parameters are determined in the similar manner for all the segments in a particular dataset.

An example of *Aloe* dataset is considered to present the output of ASC in terms of bar graph plotting and SR parameter selection. The w value for the ASC is empirically selected as 21. Initially $w = 9$ was considered for ASC but due to the lower resolution of patch it resulted in high false detections. It is observed that patch size variation does not affect the significant scale and rotation values computed for the datasets.

For each boundary patch, the best correlated patch is detected against the empirically chosen threshold value of τ_e as 3 to avoid false matches due to homogeneous regions. Figure 6.4 shows the example of scale and rotation invariant correlated patches for the *Aloe* dataset segment 1 with their corresponding SR pairs = [1, 0; 1, 1; 1.1, 0; 1, -2]. It is observed that Segment 1 contains a patterned BG which resulted in multiple SR parameter com-



(a) SR = (1, 0) (b) SR = (1, 1)



(a) SR = (1.1, 0) (b) SR = (1, -2)

Figure 6.4: Example correlated patch pairs for *Aloe* dataset segment 1 at (a) SR = (1, 0), (b) SR = (1, 1), (c) SR = (1.1, 0) and (d) SR = (1, -2).

binations whereas segment 2 mainly has homogeneous plant region and thus there is no clear scale and rotation combination other than [1, 0] i.e. no scaling or rotation. The resolution of the scale factor and rotational angle is 0.1 and 1 degree, respectively. The rotation and scale values between the correlated patches are determined by applying the Fourier Mellin Transform as in (Wilmer, 2003).

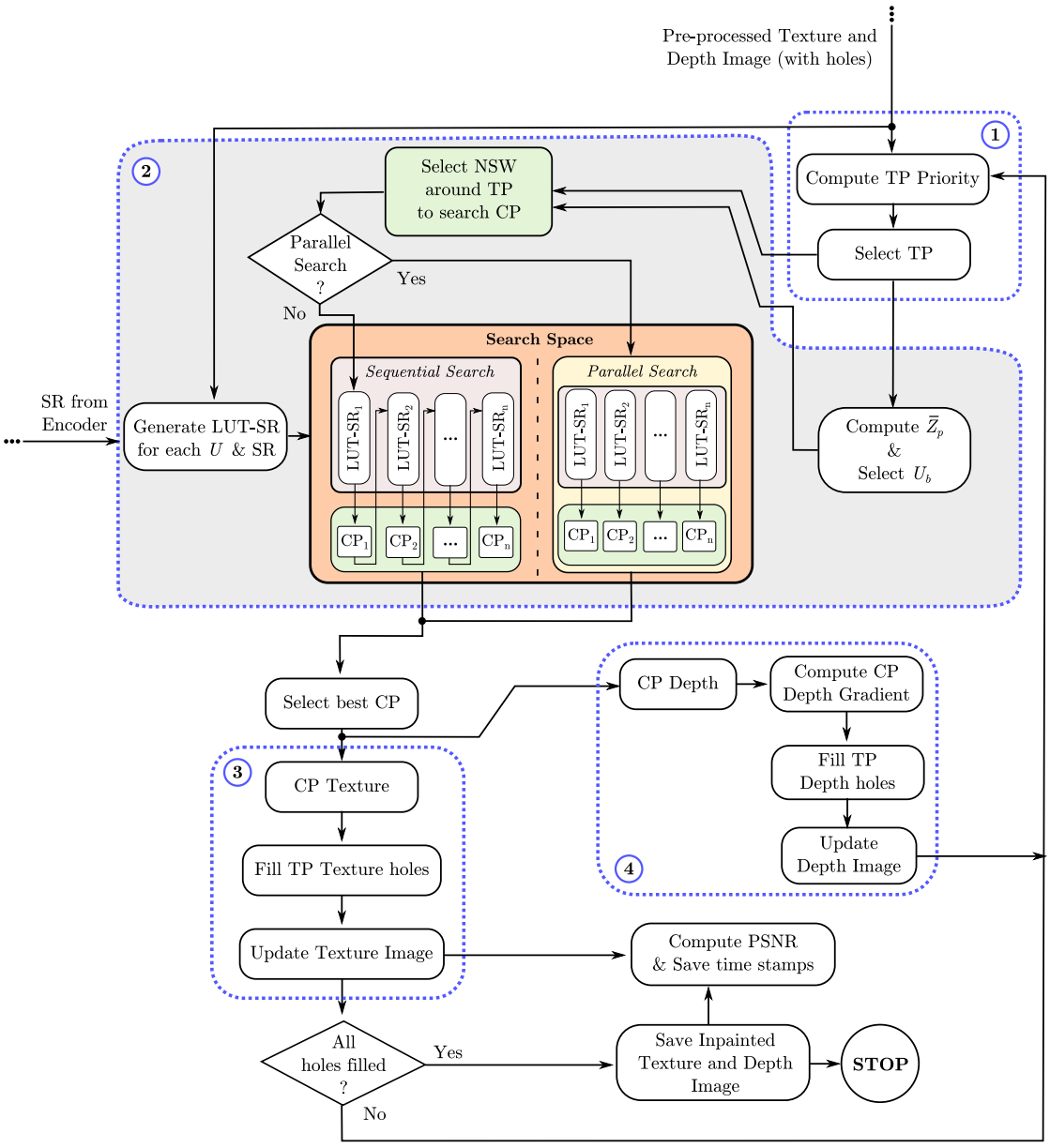


Figure 6.5: ASC-JTDI: Decoder side processing with contribution highlighted in step (2).

The selected SR parameters for each segment are sent as SI to the decoder along with the reference views and depth cut-offs. These are used to enhance the search space by generating additional candidate patches at given scales and rotation values for each segment, and will be discussed further in next section.

6.3.2 Decoder Side Processing

This section describes the steps involved in the decoder side processing for ASC-JTDI as shown in Figure 6.5. The decoder side processing is similar to Chapter 5 but differs in context of adopted search space for finding the best CP.

Step (1)**: Compute Priority**

The priority computation for selecting the patch filling order and the segment selection is similar to SC-JTDI as described earlier in step (1) in Figure 5.8. After the segment selection, the next step is to define a new NSW region.

Step (2)**: Neigbourhood Search Window Selection and Template Matching**

Unlike the exhaustive search space in JTDI, SC-JTDI narrowed the search space to a segment but included more scales to increase the availability of good patches while searching for best CP. The ASC adopts the segment based inpainting approach in SC-JTDI but confines it further to a square region, NSW of size $l \times l$ around TP as shown in Figure 6.6. This is because there is high possibility of finding a good matches near the target patch (Ashikhmin, 2001), and it provided the motivation for using NSW based segment search criteria. Such a search criteria aims to minimise the additional computation time for searching CP which increases with the addition of more parameters in generating an efficient search space while focussing on providing more accurate information around the missing pixels. As shown in the Figure 6.6, the NSW overlays on both FG and BG, how-

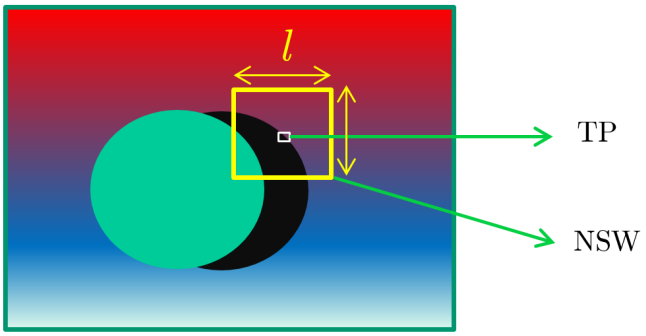


Figure 6.6: Neighbourhood Search Window around the Target Patch.

ever due to the BG segment selection in the previous step, it would only include the region which is common to both NSW and the selected segment.

The NSW is used to generate an efficient segment search space in the form of a set of Look-Up Tables (LUT) as shown in step (2) in Figure 6.5. In the search space, each $LUT - SR_s$ generated by utilising a given set of SR for each segment received from encoder, where s signifies the segment number. The TM is performed in a sequential manner to determine the best CP among all the possible candidates in each LUT-SR one at a time, and finally select the best CP among them. It is observed that with each additonal LUT-SR, such a sequential search approach becomes lengthy. Thus to minimise the search time, the TM can be transformed into a parallelised process.

MIT Lincoln Laboratory provides an excellent library called *p*MATLAB that enables parallel computing framework with MATLAB for implementing numerical computations (Kim et al., 2011). The parallel processing can be also implemented using multiple processors, Graphic Processing Unit or using MATLAB's inbuilt parallel processing toolbox. However, *p*MATLAB is freeware and its compatibility with MATLAB makes it a more viable choice over others. This technique

provides the flexibility to perform the TM through the LUT-SR either in a sequential manner or in parallel depending upon the hardware availability. The sequential approach is termed as ASC-JTDI and the parallel approach is referred to as *p*ASC-JTDI . It is used to perform parallel TM through all the LUT-SR at once and search the best CP among them. The overall aim of *p*ASC-JTDI is to minimise the candidate search time while providing the same quantitative and qualitative inpainting outcomes as ASC-JTDI.

Step (3) **and** (4)**: Texture and Depth Inpainting**

After the best CP has been selected, the joint inpainting of texture and depth holes is performed as in JTDI step (3) and (4) in Figure 4.1. The next section discusses the experimental results for ASC-JTDI.

6.4 Experimental Results and Discussion

The inpainting experiments are performed on *eight* Middlebury image datasets to evaluate the performance of ASC-JTDI. This section presents a detailed discussion on *Aloe* and *Cones,* providing their quantitative and qualitative results. The SR parameters are computed as in Section 6.3.1, followed by the inpainting process in Section 6.3.2. Firstly, the quantitative results of the inpainting are presented and thereafter the qualitative analysis is performed to show the impact of selected SR for enhanced TM. For all the experiments, $w = 9$ is considered for inpainting, as in Chapter 5 and the NSW value is empirically selected as six times the w value.

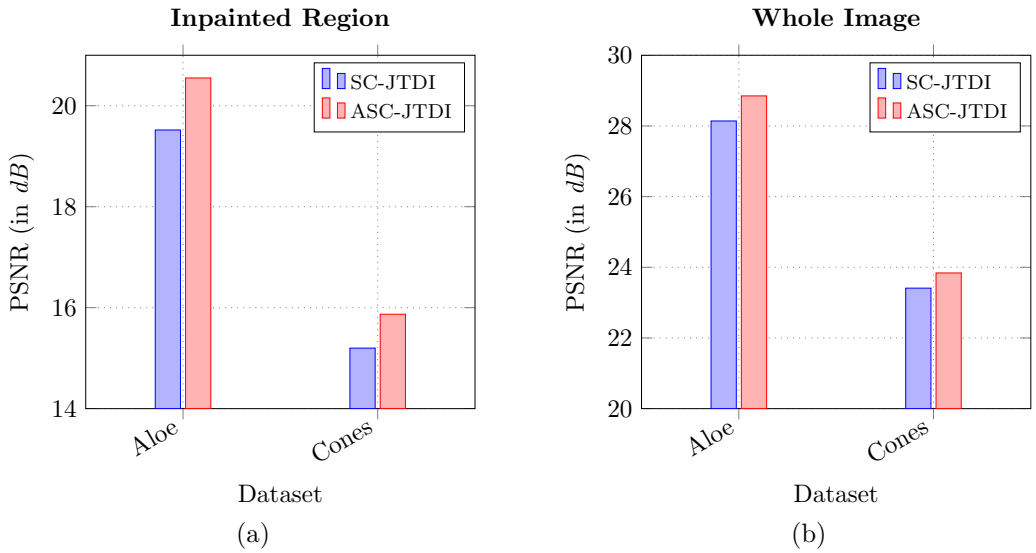


Figure 6.7: PSNR comparison for *Aloe* and *Cones* datasets in two scenarios namely, (a) Inpainted Region and (b) Whole image for ASC-JTDI and SC-JTDI respectively.

6.4.1 Quantitative Result Analysis

This section presents the quantitative performance analysis of ASC-JTDI in comparison to SC-JTDI. The PSNR analysis is performed for both 1) Whole image and 2) only the Inpainted Region. Figure 6.7 shows the PSNR results computed between the inpainted image and available ground truth image for the *Aloe* and *Cones* datasets. The plots clearly show an increase in PSNR as compared to the SC-JTDI as a result of introducing both scale and rotation parameters to enhance the search space which helped in finding superior candidate patches with lower MSE.

For *Aloe* and *Cones*, the percentage PSNR increase for inpainted region is 5.28% and 3.06% respectively, in comparison to SC-JTDI. The reason for the increased PSNR for *Aloe* is because, among the two segments, most of the holes occur in the BG segment which includes dominant SR parameters to generate the search space and results in finding good candidate patches with low MSE. Similarly, for *Cones*,

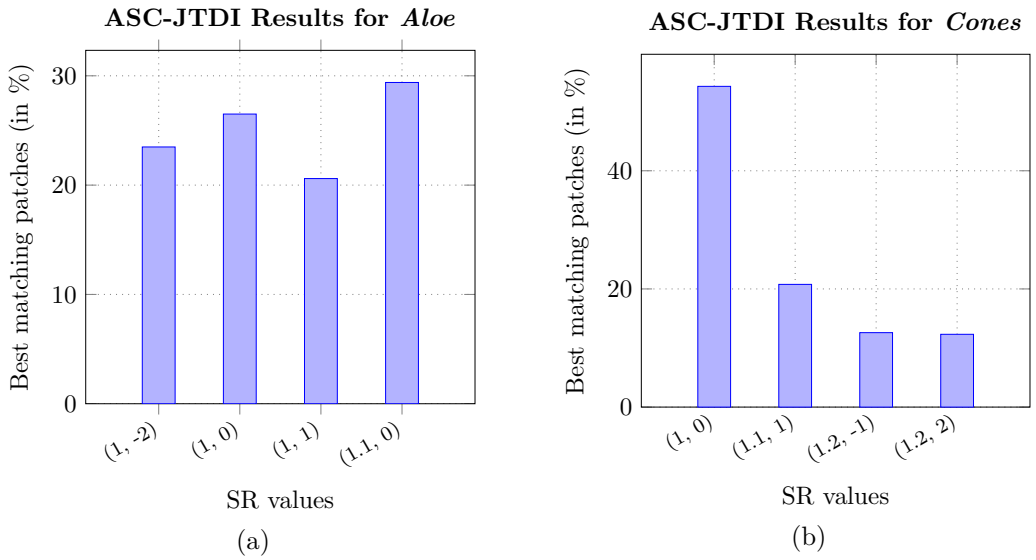


Figure 6.8: Percentage best matching patches vs SR parameters used for inpainting (a) *Aloe* (b) *Cones*.

the SR parameters are mainly associated with the BG layer but significant number of holes occurring in the in middle segment contains multiple objects and results in improved inpainting due to restricted NSW. A similar trend is also evident in other datasets, as summarised in Appendix D. However, the overall percentage increase is better in *Aloe* as compared to *Cones* which shows that the images characteristics e.g. repetitive patterns, homogeneous regions etc. impacts upon the selection of SR parameters and their contribution to the inpainting performance.

Figure 6.8 presents the impact of various SR parameters on the inpainting performance. The plot shows total number of patches inpainted using a particular SR pair which is the representation of the maximum number of patches per SR that results in lowest MSE during TM. It is observed that for *Aloe* and *Cones* dataset, four SR parameters have been used and all of them have competitively participated in providing good candidate patches during inpainting.

This provides evidence that the chosen SR parameters detected during the

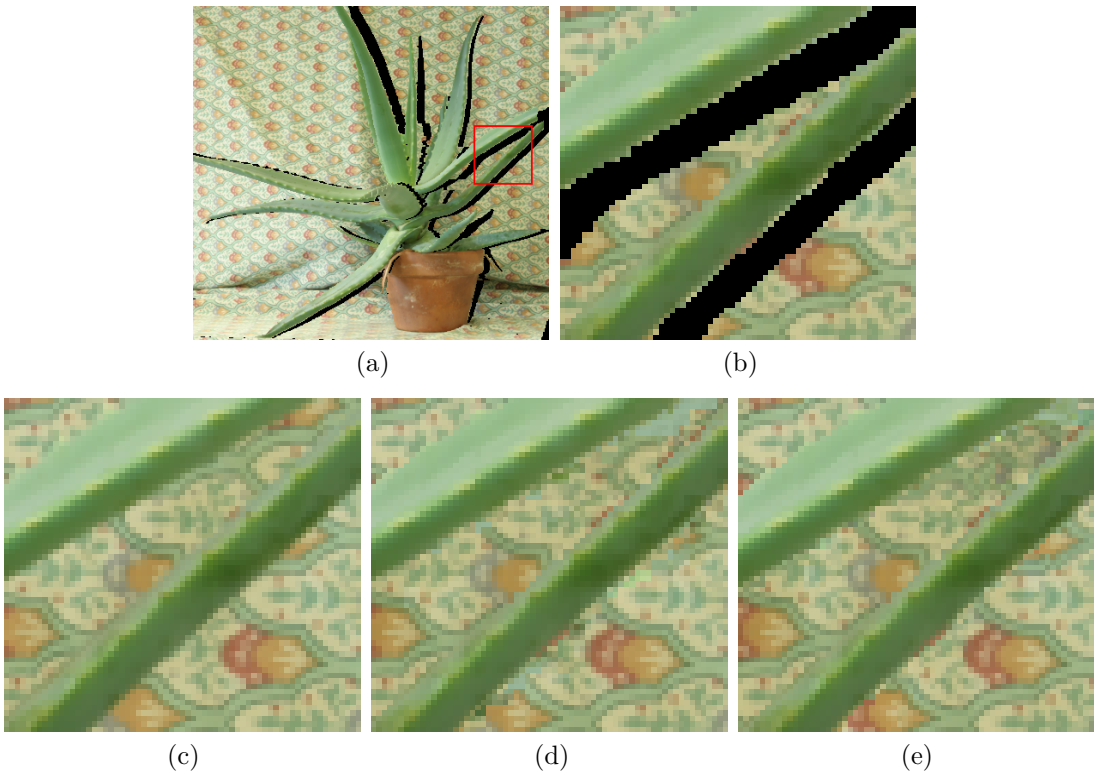


Figure 6.9: *Aloe* (a) Full Image with holes (b) Holes sub-region, (c) Ground truth, and (d) and (e) represent inpainting results by SC-JTDI and ASC-JTDI respectively.

ASC make a significant contribution to the disocclusion inpainting and results in improved quantitative performance.

6.4.2 Qualitative Result Analysis

Further to the quantitative analysis, the qualitative results are discussed for *Aloe* and *Cones*. Considering the *Aloe* dataset, Figure 6.9 represents the visual results with the zoomed-in region for the inpainted datasets to highlight upon a problem area in 6.9 (a) with its corresponding ground truth in 6.9 (c) and the inpainting results for SC-JTDI and ASC-JTDI as in Figures, 6.9 (d) and (e) respectively.

Comparing the inpainted region in Figures 6.9 (d) and (e), it is observed that ASC-JTDI shows improved inpainting around the leaf edges as a result of finding

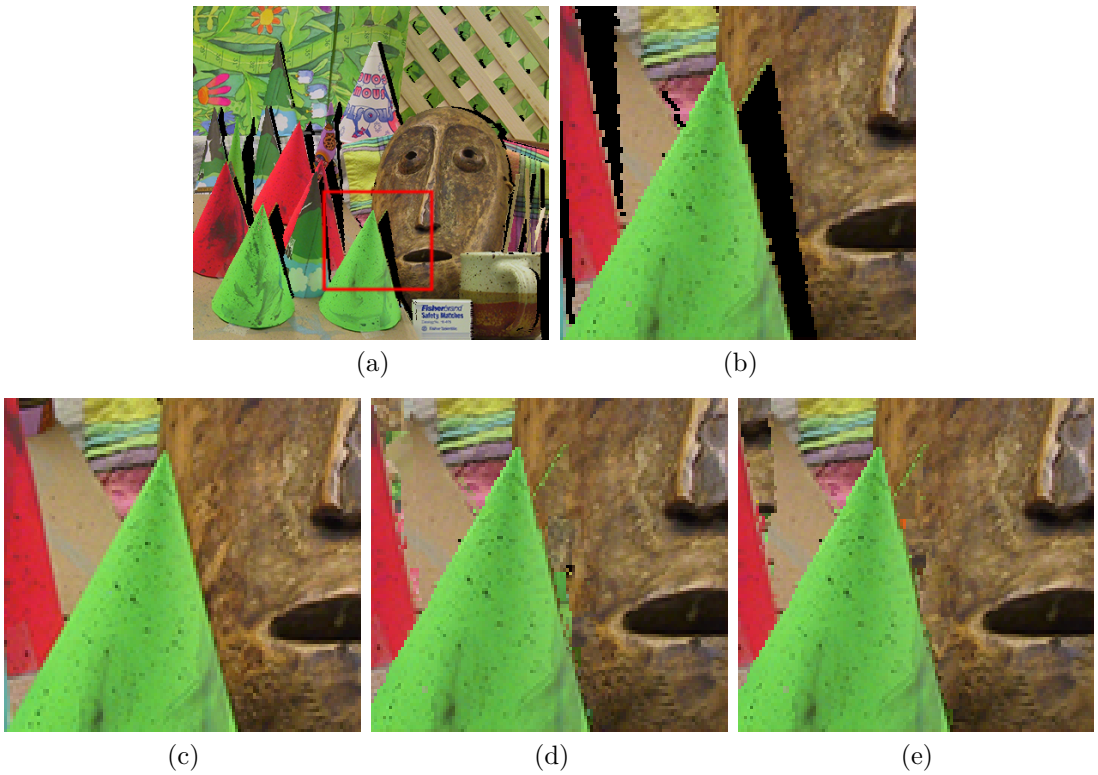


Figure 6.10: *Cones* (a) Full Image with holes (b) Holes sub-region, (c) Ground truth, and (d) and (e) represent inpainting results by SC-JTDI and ASC-JTDI respectively.

better CP against the target region in the TP, in comparison to SC-JTDI. The segmentation restricted the search space to the BG region and utilising the NSW, narrowing the search space further and providing a superior set of candidates during TM due to addition of better patches as a result of employing the SR information to generate the LUT's. The dual effect of segmentation coupled with NSW improved the overall TM and as a result, the patterned region in the BG is well-preserved and propagated to inpaint the hole region. Although the region near the top right corner is filled with the BG in (e) but it still contains artefact in comparison to the (c). This is because there exist no BG information in this region i.e. the hole region is surrounded only by the FG leaves and thus the target patches does not contain enough information to fully recover the texture.

Figure 6.10 shows the corresponding results for the *Cones* dataset. This data-

sets contains multiple overlapping objects with homogeneous regions and thus is complex to inpaint. It is observed that finding the best candidate in NSW resulted in better inpainting near the cone edge in Figure 6.10 (e) as compared to Figure 6.10 (d). This is because the NSW ignores the FG region within the search window and performs TM only in the BG patches in LUT-SR. Qualitative results for more datasets are provided in Appendix E.

Overall, it is observed that ASC-JTDI showcase better qualitative performance as compared to SC-JTDI. Both SR parameters and NSW contributed to the improved inpainting of the disocclusion regions. The next section discusses the inpainting time performance.

6.4.3 Inpainting Time Analysis

The inpainting time analysis is performed for ASC-JTDI and compared with SC-JTDI. As discussed in Section 6.3.2, the segment based search space in SC-JTDI is further confined by considering NSW in ASC-JTDI. The NSW defines the region which is used to generate a LUT by employing the SR parameters for a given segment. This aims to reduce the time involved in TM while providing superior matches by utilising enhanced search space.

For TM, the search space is generated offline and only once with a unique LUT for each set of SR parameters. For *Aloe*, 4 SR parameters have been used to generate 4 LUT's to perform TM in a serial manner to select the patch with the minimum MSE. Thus the time involved for finding a best candidate patch is four-times in comparison to if a single SR parameter LUT is used. However,

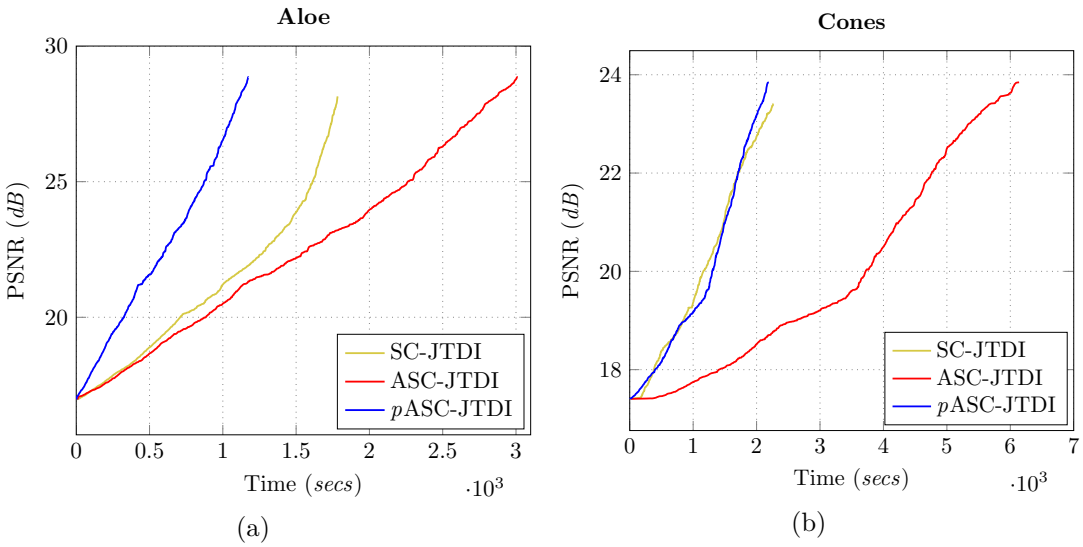


Figure 6.11: PSNR vs Time plot for SC-JTDI, ASC-JTDI and *p*ASC-JTDI with *p*MATLAB.

*p*ASC-JTDI performs parallel search in all LUT's and find the corresponding best CP in almost same time. Figure 6.11 shows *Aloe* and *Cones* plots representing PSNR vs Time analysis for *p*ASC-JTDI, ASC-JTDI and SC-JTDI.

The plot shows the ASC-JTDI has highest inpainting time followed by SC-JTDI and the lowest for *p*ASC-JTDI. This is because ASC-JTDI employs four SR parameters that are mainly associated with the BG segment which contains significant amount of holes. Thus, for inpainting each TP a sequential search through all the SR is time-consuming whereas in case of SC-JTDI, each segments has only two SP and thus it takes less time for TM in comparison to ASC-JTDI.

Although the inpainting time for SC-JTDI is less compared to ASC-JTDI, the final PSNR is also low. However, *p*ASC-JTDI presents a fair balance between time and PSNR by providing the same PSNR as ASC-JTDI but even lesser inpainting time compared to SC-JTDI. This is because the parallel TM performs simultaneous search in all the LUT's and thus minimises the overall inpainting time by almost

34% and 61% compared to SC-JTDI and ASC-JTDI respectively. The *p*ASC-JTDI shows significant improvement in inpainting time while efficiently inpainting the holes.

A similar trend is seen for *Cones* dataset, where *p*ASC-JTDI improves the inpainting time by almost 4% and 64% for SC-JTDI and ASC-JTDI respectively. It is concluded that the proposed technique provides better inpainting but at an expense of time which is efficiently minimised by parallelising the process while delivering the same output quality.

6.5 Summary

This chapter presented an Advanced Self-similarity Characterisation framework for inpainting the disocclusion holes. It exploited well-recognised LPT and FMT to automatically characterise scale and rotation invariant self-similarities within an image and utilise them for enhanced inpainting. Overall, the numerical and qualitative results for ASC-JTDI show improved and effective inpainting performance with reduced visual artefacts as compared to SC-JTDI. However, the ASC-JTDI is more effective in the presence of a patterned region than the homogeneous regions in the image, thus the inpainting performance is dependent on the image characteristics.

The flexibility of ASC-JTDI can be extended by including more image transformations for self-similarity characterisation. The increase in inpainting time due to additional characterisation parameters is compensated by using parallel imple-

mentation. This provides a proof of concept for parallel computation which can be further explored to employ GPU for real-time implementation.

Chapter 7

Future Work

The new inpainting framework presented in this thesis makes a number of original contributions to fill the disocclusion holes that appear during virtual views synthesis. There are a number of potential opportunities to extend the framework as well as to investigate extending the findings into other possible application domains. Some prospective avenues of new research building upon the findings presented in this thesis will now be discussed.

1. ASC-JTDI exploits the scale and rotation invariant, image self-similarity characteristics for inpainting disocclusion holes. As the new framework is flexible, the choice of characterisation parameters could be further extended to employ other affine transformations (Fedorov et al., 2016; Huang et al., 2014) such as shear or composite transformations in images, to enhance the candidate search space for template matching. It would also be insightful to investigate new approaches to jointly detect various transformations and employ these for inpainting holes in order to reduce visual artefacts.

2. JTDI analysed the impact of patch size on the inpainting quality and time, with the general conclusion being that a fixed patch size of 9 pixels was the best compromise. However, determining the most appropriate patch size is critical for the inpainting quality and is highly dependent on individual image characteristics. Recent exemplar-based techniques have analysed the significance of smaller and larger patch size for both structure and texture synthesis respectively (Buyssens et al., 2015). It would be beneficial to examine the feasibility of applying an adaptive patch size selection strategy tailored to the target patch characteristics. Thus a trade-off between inpainting quality and time complexity could be established by for instance, using larger patch size to fill homogeneous regions and smaller patches for highly textured regions such as holes between multiple FG objects.

3. ASC-JTDI flexibly chose the scale and rotation parameters for a given image, however, the computational complexity increase with the number of parameters selected for search space enhancement. To overcome the additional complexity, the search was confined to neighbourhood region and further, employing *p*MATLAB has shown to improve inpainting times. Investigating techniques to speed-up the inpainting by adopting a faster simulation platform such as *Graphic Processing Unit* would be beneficial for real-time inpainting applications (Kuo et al., 2013, 2015) in FVV.

4. 3D point cloud data captured by *Light Detection and Ranging* (LiDAR) systems for surveying and architectural applications often contain large numbers of texture holes behind FG objects of nearly all real-world scans (Doria and Radke, 2012; Kobal et al., 2015). Synthesising realistic information in these

large holes would represent a challenging extension for the new inpainting framework. Colour images captured by *Kinect* cameras often contain holes in their corresponding depth maps due to occlusions, transparent objects or scattering. Filling these holes effectively is another possible future research direction (Hu et al., 2013; Wang et al., 2014).

5. While inpainting has significantly improved the visual quality and proved its worth in various multimedia applications, it is also gaining attention in other domains like medical imaging and remote sensing. For example, it has been used to reduce the impact of undesired features by pre-processing intravascular ultrasound (IVUS) images (Stolojescu-Crisan and Isar11, 2015) and reducing CT metal artefacts by inpainting sinogram (Chen et al., 2012). Inpainting also finds it applicability in compressive sensing (Stolojescu-Crisan and Isar, 2015) and remote sensing (Cerra et al., 2015). It is promising to investigate some of the unique challenges in these various application domains in order to evaluate how the new inpainting framework can be advanced or refined to support the essential robust inpainting strategies required.

Chapter 8

Conclusion

Advances in multimedia technologies have inspired considerable research into interactive multi-view applications like free viewpoint video, with the aim of providing users with an immersive experience by allowing free navigation between views, without confining the viewer to only broadcasted views. While Depth Image-Based Rendering enables the synthesis of arbitrary virtual views from the available set of transmitted views, these almost inevitably include disocclusion holes which must be filled to achieve a visually pleasing virtual image. Traditional 2D inpainting methods employ purely textural information which is inadequate for disocclusion hole-filling and has led to depth-assisted inpainting solutions being developed to address the challenging hole-filling problem. The most prominent disocclusion holes inpainting methods are exemplar-based, which utilise spatial information from reference views, but these often do not provide sufficient numbers of good candidate patches for effective template matching. This was the motivation behind the research question to investigate novel inpainting approaches to achieve

perceptually pleasing virtual view synthesis.

This thesis has presented a new inpainting framework for virtual view synthesis which efficiently uses a depth-assisted solution to uniquely exploit image transformational self-similarities and joint texture and depth inpainting of disocclusion holes. The new framework makes three original contributions to the field:

1. The most significant is the Advanced Self-similarity Characterisation based Joint Texture-Depth Inpainting which automatically determines the key scale and rotational parameters in a reference image to jointly inpaint disocclusion holes in both the texture and depth maps of the virtual view. The approach characterises scale and rotation invariant self-similarities to determine the dominant scales and rotation values for each segment of the image and applies them to generate a rich search space for candidate selection whilst confining it to a narrow search window for efficient inpainting. The approach is flexible so it can be extended to include the characterisation of additional image self-similarity features and can reduce the inpainting time by adopting parallel programming techniques. Experimental results conclusively show the superior inpainting performance achieved with this framework, with for example, a PSNR gain of 25.22% for the *Aloe* dataset.

2. Underpinning the first contribution is the introduction of the concept of Self-similarity Characterisation to enhance inpainting performance. The original self-similarity based Joint Texture-Depth Inpainting technique employed a characterisation mechanism utilising an empirically selected scale range for segment-based, multi-scale self-similarity analysis at the encoder. The result-

ing transmitted scaling parameters enrich the segment-based search space at decoder for simultaneous hole-filling. This encoder-guided approach requires only one self-similarity characterisation at the encoder. This has the benefit of avoiding the imposition of additional complexity upon the decoder, while securing a superior search space for fast template matching. The segment-based inpainting approach improves the selection of candidate patches to reduce the resulting perceptual artefacts during the hole-filling process.

3. The original Joint Texture-Depth Inpainting algorithm is the core constituent block of all the presented inpainting contributions of the new framework, focusing explicitly on the joint inpainting of texture and depth virtual views. It uses available depth information to guide texture hole-filling and then utilised the in-filled texture information to assist in-filling the depth holes. A new depth oriented priority term ensured an effective filling order to minimise error propagation during inpainting, while empirical patch size evaluations provided the design flexibility to trade between inpainting speed and quality. In comparison to existing inpainting techniques, this technique produced superior and more robust performance under a variety of test datasets.

In reflecting on the main features and performance benefits of the new inpainting framework and contrasting with existing schemes, it presents an innovative solution for effective and efficient disocclusion hole inpainting in terms of reduced perceptual artefacts. From a practical perspective, it is recognised that many issues remain to be resolved in regard to how accurate, real-time inpainting can be achieved in applications like FVV. However, overall, the new framework makes a notable contribution to the inpainting field by affording both a robust and extend-

able platform on which to develop real-world inpainting solutions for virtual view synthesis.

References

Ahn, I. and Kim, C. (2012). Depth-based disocclusion filling for virtual view synthesis. In *2012 IEEE International Conference on Multimedia and Expo (ICME)*, pages 109 –114.

Ahn, I. and Kim, C. (2013). A novel depth-based virtual view synthesis method for free viewpoint video. *IEEE Transactions on Broadcasting*, 59(4):614–626.

Araujo, H. and Dias, J. M. (1996). An introduction to the log-polar mapping [image sampling]. In *, Second Workshop on Cybernetic Vision, 1996. Proceedings*, pages 139–144.

Arias, P., Caselles, V., and Sapiro, G. (2009). A variational framework for non-local image inpainting. In Cremers, D., Boykov, Y., Blake, A., and Schmidt, F. R., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, number 5681 in Lecture Notes in Computer Science, pages 345–358. Springer Berlin Heidelberg.

Ashikhmin, M. (2001). Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, I3D '01, pages 217–226, New York, NY, USA. ACM.

Aujol, J., Ladjal, S., and Masnou, S. (2010). Exemplar-based inpainting from a variational point of view. *SIAM Journal on Mathematical Analysis*, 42(3):1246–1285.

Azzari, L., Battisti, F., and Gotchev, A. (2010). Comparative analysis of occlusion-filling techniques in depth image-based rendering for 3d videos. pages 57–62. ACM Press.

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., and Verdera, J. (2001a). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200 –1211.

Ballester, C., Caselles, V., Verdera, J., Bertalmio, M., and Sapiro, G. (2001b). A variational model for filling-in gray level and color images. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, volume 1, pages 10 –16.

Barnes, C., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2010). The generalized patchmatch correspondence algorithm. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision ECCV 2010*, number 6313 in Lecture Notes in Computer Science, pages 29–43. Springer Berlin Heidelberg.

Bertalmio, M. (2001). *Processing of flat and non-flat image information on arbitrary manifolds using partial differential equations.* PhD thesis, Computer Eng. Program.

Bertalmio, M., Bertozzi, A., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–355 – I–362.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques SIGGRAPH 00*, 2(5):417–424.

Bhat, S. (n.d.). Object removal by exemplar-based inpainting. [Online]. Available at http://www.cc.gatech.edu/ šooraj/inpainting/ (Accessed 11th December 2014).

Bornard, R., Lecan, E., Laborelli, L., and Chenot, J.-H. (2002). Missing data correction in still images and image sequences. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 355–361, New York, NY, USA. ACM.

Bozek, P. and Pivarciova, E. (2012). Registration of holographic images based on integral transformation. 31(6):1369–1383.

Bracewell, R. N. (1999). *The Fourier Transform & Its Applications*. McGraw-Hill Higher Education, Boston, 3 edition.

Bravo-Solorio, S. and Nandi, A. K. (2011). Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics. *Signal Processing*, 91(8):1759–1770.

Buades, A., Coll, B., and Morel, J. M. (2005). A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65.

Bugeau, A., Bertalmilo, M., Caselles, V., and Sapiro, G. (2010). A comprehensive framework for image inpainting. *IEEE Transactions on Image Processing*, 19(10):2634 –2645.

Buyssens, P., Daisy, M., Tschumperle, D., and Lezoray, O. (2015). Exemplar-Based Inpainting: technical review and new heuristics for better geometric reconstructions. *IEEE Transactions on Image Processing*, 24(6):1809–1824.

Cao, F., Gousseau, Y., Masnou, S., and Perez, P. (2011). Geometrically guided exemplar-based inpainting. *SIAM J. Img. Sci.*, 4(4):1143–1179.

Cerra, D., Bieniarz, J., Maeller, R., Storch, T., and Reinartz, P. (2015). Restoration of Simulated EnMAP Data through Sparse Spectral Unmixing. *Remote Sensing*, 7(10):13190–13207.

Chan, T. and Shen, J. (2001). Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math*, 62:1019–1043.

Chan, T. F., Kang, S. H., Kang, and Shen, J. (2002). Euler's elastica and curvature based inpaintings. *SIAM J. Appl. Math*, 63:564–592.

Chen, K.-Y., Tsung, P.-K., Lin, P.-C., Yang, H.-J., and Chen, L.-G. (2010a). Hybrid motion/depth-oriented inpainting for virtual view synthesis in multiview applications. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1 –4.

Chen, Q.-S., Defrise, M., and Deconinck, F. (1994). Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1156–1168.

Chen, W.-Y., Chang, Y.-L., Lin, S.-F., Ding, L.-F., and Chen, L.-G. (2005). Efficient depth image based rendering with edge dependent depth filter and interpolation. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1314–1317.

Chen, Y., Li, Y., Guo, H., Hu, Y., Luo, L., Yin, X., Gu, J., and Toumoulin, C. (2012). CT Metal artifact reduction method based on improved image segmentation and sinogram In-painting, CT metal artifact reduction method based on improved image segmentation and sinogram In-painting. *Mathematical Problems in Engineering, Mathematical Problems in Engineering*, page e786281.

Chen, Y., Wan, W., Hannuksela, M. M., Zhang, J., Li, H., and Gabbouj, M. (2010b). Depth-level-adaptive view synthesis for 3d video. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1724–1729.

Cheng, C. M., Lin, S. J., and Lai, S. H. (2011). Spatio-temporally consistent novel view synthesis algorithm from video-plus-depth sequences for autostereoscopic displays. *IEEE Transactions on Broadcasting*, 57(2):523–532.

Cheng, C.-M., Lin, S.-J., Lai, S.-H., and Yang, J.-C. (2008). Improved novel view synthesis from depth image with large baseline. In *19th International Conference on Pattern Recognition, 2008. ICPR 2008*, pages 1 –4.

Cheung, C. H., Sheng, L., and Ngan, K. N. (2015). A disocclusion filling method using multiple sprites with depth for virtual view synthesis. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6.

Crary, J. (1992). *Techniques of the observer: on vision and modernity in the nineteenth century*. MIT Press.

Criminisi, A., Perez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200 –1212.

Daribo, I., Cheung, G., Maugey, T., and Frossard, P. (2012). R-D optimized auxiliary information for inpainting-based view synthesis. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pages 1 –4.

Daribo, I. and Pesquet-Popescu, B. (2010). Depth-aided image inpainting for novel view synthesis. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 167–170.

Daribo, I. and Saito, H. (2011). A novel inpainting-based layered depth video for 3dtv. *IEEE Transactions on Broadcasting*, 57(2):533 –541.

Daubechies, I., DeVore, R., Fornasier, M., and Gunturk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38.

De Bonet, J. S. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 361–368, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Ding, T., Sznaier, M., and Camps, O. I. (2007). A rank minimization approach to video inpainting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

Do, L., Zinger, S., Morvan, Y., and de With, P. (2009). Quality improving techniques in DIBR for free-viewpoint video. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1 –4.

Domaski, M., Gotfryd, M., and Wegner, K. (2009). View synthesis for multiview video transmission. In *he 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition IPCV*, volume 9, Las Vegas.

Doria, D. and Radke, R. J. (2012). Filling large holes in lidar data by inpainting depth gradients. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 65–72.

Drori, I., Cohen-Or, D., and Yeshurun, H. (2003). Fragment-based image completion. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 303–312, New York, NY, USA. ACM.

Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*, volume 2, pages 1033 –1038.

Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 341–346, New York, NY, USA. ACM.

Elad, M., Starck, J.-L., Querre, P., and Donoho, D. (2005). Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340–358.

Emori, T., Tehrani, M. P., Takahashi, K., and Fujii, T. (2015). Free-viewpoint video synthesis from mixed resolution multi-view images and low resolution depth maps. volume 9391, pages 93911C–93911C–10.

Fang, L., Cheung, N. M., Tian, D., Vetro, A., Sun, H., and Au, O. C. (2014). An analytical model for synthesis distortion estimation in 3d video. *IEEE Transactions on Image Processing*, 23(1):185–199.

Farid, M. S., Lucenteforte, M., and Grangetto, M. (2014). Edge enhancement of depth based rendered images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5452–5456.

Fedorov, V., Arias, P., Facciolo, G., and Ballester, C. (2016). Affine invariant self-similarity for exemplar-based inpainting. In *VISAPP*.

Fedorov, V., Arias, P., Sadek, R., Facciolo, G., and Ballester, C. (2015). Linear multiscale analysis of similarities between images on riemannian manifolds: practical formula and affine covariant metrics. *SIAM Journal on Imaging Sciences*, 8(3):2021–2069.

Fehn, C. (2004a). Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3d-TV. pages 93–104.

Fehn, C. (2004b). Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. *Proc. SPIE*, 5291:93–104.

Fujii, T. and Tanimoto, M. (2002). Free viewpoint TV system based on ray-space representation. pages 175–189.

Gao, Y., Chen, H., Gao, W., and Vaudrey, T. (2013). Virtual view synthesis based on DIBR and image inpainting. In Klette, R., Rivera, M., and Satoh, S., editors, *Image and Video Technology*, number 8333 in Lecture Notes in Computer Science, pages 172–183. Springer Berlin Heidelberg.

Gautier, J., Le Meur, O., and Guillemot, C. (2011). Depth-based image completion for view synthesis. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1 –4.

Girod, B. (1991). Psychovisual aspects Of image processing: What's wrong with mean squared error? In *, Proceedings of the Seventh Workshop on Multidimensional Signal Processing, 1991*, pages P.2–P.2.

Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall.

Gui, H., Pang, Z., Chen, D., Chen, M., and Tan, H. (2013). A forward and reverse wrapping depth image-based rendering (FR-DIBR) method for arbitrary view generation. In Yang, Y. and Ma, M., editors, *Proceedings of the 2nd International Conference on Green Communications and Networks 2012 (GCN 2012): Volume 1*, number 223 in Lecture Notes in Electrical Engineering, pages 683–690. Springer Berlin Heidelberg.

Guillemot, C. and Meur, O. L. (2014). Image Inpainting : Overview and recent advances. *IEEE Signal Processing Magazine*, 31(1):127–144.

Heeger, D. and Bergen, J. (1995). Pyramid-based texture analysis/synthesis. In *, International Conference on Image Processing, 1995. Proceedings*, volume 3, pages 648 –651.

Hirschmuller, H. and Scharstein, D. (2007). Evaluation of cost functions for stereo matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Horaud, R., Hansard, M., Evangelidis, G., and Ménier, C. (2016). An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine Vision and Applications*, 27(7):1005–1020.

Hornung, A. and Kobbelt, L. (2009). Interactive pixel-accurate free viewpoint rendering from images with silhouette aware sampling. *Computer Graphics Forum*, 28(8):2090–2103.

Hu, J., Hu, R., Wang, Z., Gong, Y., and Duan, M. (2013). Color image guided locality regularized representation for Kinect depth holes filling. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6.

Huang, J.-B., Kang, S. B., Ahuja, N., and Kopf, J. (2014). Image completion using planar structure guidance. *ACM Trans. Graph.*, 33(4):129:1–129:10.

Jain, A., Tran, L., Khoshabeh, R., and Nguyen, T. (2011). Efficient stereo-to-multiview synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*, pages 889 –892.

Jantet, V., Guillemot, C., and Morin, L. (2011). Joint projection filling method for occlusion handling in Depth-Image-Based Rendering. *3D Research*, 2(4):1–13.

Jiufei, X., Ming, X., Dongxiao, L., and Ming, Z. (2010). A new virtual view rendering method based on depth image. In *A new virtual view rendering method based on depth image. In Wearable Computing Systems (APWCS), 2010 Asia-Pacific Conference*, pages 147 –150.

Jurie, F. (1999). A new log-polar mapping for space variant imaging. *Pattern Recognition*, 32(5):865–875.

Kauff, P., Atzpadin, N., Fehn, C., Maller, M., Schreer, O., Smolic, A., and Tanger, R. (2007). Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. *Signal Processing: Image Communication*, 22(2):217–234.

Kawai, N., Sato, T., and Yokoya, N. (2009). Image inpainting considering brightness change and spatial locality of textures and Its evaluation. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Wada, T., Huang, F., and Lin, S., editors, *Advances in Image and Video Technology*, volume 5414, pages 271–282. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kim, H., Mullen, J., and Kepner, J. (2011). Introduction to parallel programming and pMatlab v2. 0. Technical report, DTIC Document.

Kobal, M., Bertoncelj, I., Pirotti, F., Dakskobler, I., and Kutnar, L. (2015). Using Lidar data to analyse sinkhole characteristics relevant for understory vegetation under forest covercase study of a high Karst area in the dinaric mountains. *PLoS ONE*, 10(3).

Kohli, P., Rother, C. C. E., and Sharp, T. (2012). Image completion using scene geometry. International Classification: G06K9/36; G06K9/00.

Kondoz, A. and Dagiuklas, T. (2013). *3D Future Internet Media*. Springer Science & Business Media.

Kondoz, A. and Dagiuklas, T. (2014). *Novel 3D Media Technologies*. Springer.

Koppel, M., Ndjiki-Nya, P., Doshkov, D., Lakshman, H., Merkle, P., Muller, K., and Wiegand, T. (2010). Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering. In *2010 17th IEEE International Conference on Image Processing (ICIP)*, pages 1809 –1812.

Kryszkiewicz, M., Bandyopadhyay, S., Rybinski, H., and Pal, S. K. (2015). *Pattern Recognition and Machine Intelligence: 6th International Conference, PReMI 2015, Warsaw, Poland, June 30 - July 3, 2015, Proceedings*. Springer.

Kubota, A., Smolic, A., Magnor, M., Tanimoto, M., Chen, T., and Zhang, C. (2007). Multiview Imaging and 3dtv. *Signal Processing Magazine, IEEE*, 24(6):10 –21.

Kuo, P.-C., Lin, J.-M., Liu, B.-D., and Yang, J. F. (2013). Inpainting-based multiview synthesis algorithms and its GPU accelerated implementation. In *Communications and Signal Processing (ICICS) 2013 9th International Conference on Information*, pages 1–4.

Kuo, P.-C., Lin, J.-M., Liu, B.-D., and Yang, J.-F. (2015). High efficiency depth image-based rendering with simplified inpainting-based hole filling. *Multidimensional Systems and Signal Processing*, 27(3):623–645.

Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. (2003). Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286.

Lan, C., Xu, J., Wu, F., and Shi, G. (2010). Intra frame coding with template matching prediction and adaptive transform. In *2010 IEEE International Conference on Image Processing*, pages 1221–1224.

Lefebvre, S. and Hoppe, H. (2006). Appearance-space texture synthesis. SIGGRAPH '06, pages 541–548, New York, NY, USA. ACM.

Lei, J., Zhang, C., Wu, M., You, L., Fan, K., and Hou, C. (2016). A divide-and-conquer hole-filling method for handling disocclusion in single-view rendering. *Multimedia Tools and Applications*, pages 1–16.

Leuven, K. (n.d.). Structured light. [Online]. Available at http://www.esat.kuleuven.be/psi/research/structured-light (Accessed 1st May 2017).

Lin, C. Y., Wu, M., Bloom, J. A., Cox, I. J., Miller, M. L., and Lui, Y. M. (2001). Rotation, scale, and translation resilient watermarking for images. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 10(5):767–782.

Lu, X.-h., Wei, F., and Chen, F.-m. (2012). Foreground-object-protected depth map smoothing for DIBR. pages 339 –343.

Lukac, R. (2012). *Perceptual Digital Imaging: Methods and Applications*. CRC Press.

Ma, L., Do, L., and de With, P. H. N. (2012). Depth-guided inpainting algorithm for Free-Viewpoint Video. In *2012 19th IEEE International Conference on Image Processing (ICIP)*, pages 1721–1724.

Mahersia, H. and Hamrouni, K. (2008). New rotaion invariant features for texture classification. In *International Conference on Computer and Communication Engineering, 2008. ICCCE 2008*, pages 687–690.

Mairal, J., Elad, M., and Sapiro, G. (2008a). Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53 –69.

Mairal, J., Sapiro, G., and Elad, M. (2008b). Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241.

Majumdar, A. and Ward, R. (2011). Some empirical advances in matrix completion. *Signal Processing*, 91(5):1334–1338.

Majumdar, A., Ward, R. K., and Aboulnasr, T. (2012). A focuss based method for low rank matrix recovery. In *2012 19th IEEE International Conference on Image Processing (ICIP)*, pages 1713–1716.

Mansfield, A., Prasad, M., Rother, C., Sharp, T., Kohli, P., and Gool, L. V. (2011). Transforming image completion. pages 121.1–121.11. British Machine Vision Association.

Mark, W. (1999). Post-rendering 3d image warping: Visibility, reconstruction, and performance for depth-image warping. Technical report, Chapel Hill, NC, USA.

Mark, W. R., McMillan, L., and Bishop, G. (1997). Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 7–ff., New York, NY, USA. ACM.

Martanez-Noriega, R., Roumy, A., and Blanchard, G. (2012). Exemplar-based image inpainting: Fast priority and coherent nearest neighbor search. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6.

Martanez-Rach, M. O., Piaol, P., Lapez, O. M., Perez Malumbres, M., Oliver, J., and Calafate, C. T. (2014). On the performance of video quality assessment metrics under different compression and packet loss scenarios. *The Scientific World Journal*, 2014:1–18.

Masnou, S. and Morel, J.-M. (1998). Level lines based disocclusion. In *1998 International Conference on Image Processing, 1998. ICIP 98. Proceedings*, pages 259 –263 vol.3.

Mathworks (n.d.). Matlab - mathworks united kingdom. [Online]. Available at http://uk.mathworks.com/ (Accessed 1st August 2014).

Matungka, R. (2009). *Studies on Log-Polar Transform for Image Registration and Improvements Using Adaptive Sampling and Logarithmic Spiral*. PhD thesis, The Ohio State University.

McMillan, Jr., L. (1997). *An image-based approach to three-dimensional computer graphics*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

Merkle, P., Morvan, Y., Smolic, A., Farin, D., Malller, K., de With, P. H. N., and Wiegand, T. (2009). The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communication*, 24:73–88.

Merkle, P., Smolic, A., Muller, K., and Wiegand, T. (2007). Multi-view video plus depth representation and coding. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I – 201–I – 204.

Meur, O. L., Gautier, J., and Guillemot, C. (2011). Examplar-based inpainting based on local geometry. In *2011 18th IEEE International Conference on Image Processing*, pages 3401–3404.

Mohan, K. and Fazel, M. (2010). Reweighted nuclear norm minimization with application to system identification. In *American Control Conference (ACC), 2010*, pages 2953 –2959.

Mohan, K. and Fazel, M. (2012). Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13:3441–3473.

Mori, Y., Fukushima, N., Fujii, T., and Tanimoto, M. (2008). View generation with 3d warping using depth information for FTV. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pages 229 –232.

Morvan, Y., Farin, D., and de With, P. (2008). System architecture for free-viewpoint video and 3d-TV. *IEEE Transactions on Consumer Electronics*, 54(2):925–932.

Muddala, S. M., Olsson, R., and Sojostrom, M. (2013). Disocclusion handling using depth-based inpainting. pages 136–141.

Muddala, S. M., Olsson, R., and Sojostrom, M. (2016). Spatio-temporal consistent depth-image-based rendering using layered depth image and inpainting. *EURASIP Journal on Image and Video Processing*, 2016(1).

Muller, K., Merkle, P., and Wiegand, T. (2011). 3-D video representation using depth maps. *Proceedings of the IEEE*, 99(4):643–656.

Muller, K., Smolic, A., Dix, K., Kauff, P., and Wiegand, T. (2008a). Reliability-based generation and view synthesis in layered depth video. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 34 –39.

Muller, K., Smolic, A., Dix, K., Merkle, P., Kauff, P., and Wiegand, T. (2008b). View synthesis for advanced 3d video systems. *EURASIP Journal on Image and Video Processing*, 2008:1–11.

Myna, A. N., Venkateshmurthy, M. G., and Patil, C. G. (2007). Detection of region duplication forgery in digital images using wavelets and log-polar mapping. In *International Conference on Conference on Computational Intelligence and Multimedia Applications, 2007*, volume 3, pages 371–377.

Ndjiki-Nya, P., Kappel, M., Doshkov, D., Lakshman, H., Merkle, P., Maller, K., and Wiegand, T. (2010). Depth image based rendering with advanced texture synthesis. In *2010 IEEE International Conference on Multimedia and Expo (ICME)*, pages 424–429.

Ndjiki-Nya, P., Koppel, M., Doshkov, D., Lakshman, H., Merkle, P., Muller, K., and Wiegand, T. (2011). Depth image-based rendering with advanced texture synthesis for 3-D video. *IEEE Transactions on Multimedia*, 13(3):453–465.

Ndjiki-Nya, P., Koppel, M., Doshkov, D., and Wiegand, T. (2008). Automatic Structure-Aware Inpainting for Complex Image Content. ISVC '08, pages 1144–1156, Berlin, Heidelberg. Springer-Verlag.

Nie, D., Ma, L., and Xiao, S. (2006). Similarity based image inpainting method. In *Multi-Media Modelling Conference Proceedings, 2006 12th International*, page 4 pp.

Oh, K.-J., Yea, S., and Ho, Y.-S. (2009). Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1 –4.

Oliveira, M. M., Bowen, B., Mckenna, R., and Chang, Y.-s. (2001). Fast digital image inpainting. pages 261–266. ACTA Press.

Oncu, A. I., Deger, F., and Hardeberg, J. Y. (2012). Evaluation of digital inpainting quality in the context of artwork restoration. In *Proceedings of the 12th International Conference on Computer Vision - Volume Part I*, ECCV'12, pages 561–570, Berlin, Heidelberg. Springer-Verlag.

Ono, S., Miyata, T., Yamada, I., and Yamaoka, K. (2012). Missing region recovery by promoting blockwise low-rankness. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1281 –1284.

Owens, J. (2016). *Television Production*. CRC Press.

Panigrahi, N. (2014). *Computing in Geographic Information Systems*. CRC Press.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70.

Ramachandran, G. and Rupp, M. (2012). Multiview synthesis from stereo views. In *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 341–345.

Raman, S. and Desai, U. (1995). 2-D object recognition using Fourier Mellin transform and a MLP network. volume 4, pages 2154–2156. IEEE.

Reddy, B. S. and Chatterji, B. N. (1996). An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271.

Reel, S., Cheung, G., Wong, P., and Dooley, L. S. (2013). Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–7.

Reel, S., Wong, K. C. P., Cheung, G., and Dooley, L. S. (2014). Disocclusion hole-filling in dibr-synthesized images using multi-scale template matching. In *Visual Communications and Image Processing Conference, 2014 IEEE*, pages 494–497.

Sarvaiya, J. N., Patnaik, S., and Bombaywala, S. (2009). Image registration using log-polar transform and phase correlation. In *TENCON 2009 - 2009 IEEE Region 10 Conference*, pages 1–5.

Scharstein, D., Hirschmuller, H., Kitajima, Y., Krathwohl, G., Ne, N., Wang, X., and Westling, P. (2014). High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In Jiang, X., Hornegger, J., and Koch, R., editors, *Pattern Recognition*, volume 8753, pages 31–42. Springer International Publishing, Cham.

Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, pages 1 –8.

Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, volume 1, pages I–195 – I–202.

Schmeing, M. and Jiang, X. (2015). Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting. *IEEE Transactions on Multimedia*, 17(12):2160–2173.

Schreer, O., Kauff, P., and Sikora, T. (2005). *3D Videocommunication: Algorithms, Concepts and Real-time Systems in Human Centred Communication*. Wiley.

Seshadrinathan, K., Soundararajan, R., Bovik, A. C., and Cormack, L. K. (2010). Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing*, 19(6):1427–1441.

Shade, J., Gortler, S., He, L.-w., and Szeliski, R. (1998). Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA. ACM.

Shen, B., Hu, W., Zhang, Y., and Zhang, Y.-J. (2009). Image inpainting via sparse representation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, pages 697 –700.

Silva, D. V. S. X. D., Fernando, W. A. C., Kodikaraarachchi, H., Worrall, S. T., and Kondoz, A. M. (2010). Adaptive sharpening of depth maps for 3d-TV. *Electronics Letters*, 46(23):1546–1548.

Singh, S., Singh, M., Apte, C., and Perner, P. (2005). *Pattern Recognition and Image Analysis: Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005.* Springer.

Smolic, A. and Kauff, P. (2005). Interactive 3-D video representation and coding technologies. *Proceedings of the IEEE*, 93(1):98 –110.

Smolic, A., Kauff, P., Knorr, S., Hornung, A., Kunter, M., Muller, M., and Lang, M. (2011). Three-dimensional video postproduction and processing. *Proceedings of the IEEE*, 99(4):607 –625.

Smolic, A., Muller, K., Dix, K., Merkle, P., Kauff, P., and Wiegand, T. (2008). Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems. In *15th IEEE International Conference on Image Processing, 2008. ICIP 2008*, pages 2448 –2451.

Solh, M. and AlRegib, G. (2012). Hierarchical hole-filling for depth-based view synthesis in FTV and 3d video. *Selected Topics in Signal Processing, IEEE Journal of*, PP(99):1.

Stemmer Imaging Ltd. (n.d.). 3d time of flight cameras. [Online]. Available at https://www.stemmer-imaging.co.uk/en/knowledge-base/cameras-3d-time-of-flight-cameras/ (Accessed 1st May 2017).

Stolojescu-Crisan, C. and Isar, A. (2015). Images compressive sensing reconstruction by inpainting. In *2015 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4.

Stolojescu-Crisan, C. and Isar11, A. (2015). Ultrasound images conditioning. *Revue Roumaine Des Sciences Techniques-Serie Electechnique Et Energetique*, 60(3):303–312.

Stone, M. (2016). *A Field Guide to Digital Color*. CRC Press.

Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. (2005). Image completion with structure propagation. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 861–868, New York, NY, USA. ACM.

Sznaier, M. and Camps, O. (2005). A Hankel operator approach to texture modelling and inpainting. In *Texture 2005 : Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pages 125–130.

Takahashi, T., Konishi, K., and Furukawa, T. (2011). Reweighted l2 norm minimization approach to image inpainting based on rank minimization. In *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1 –4.

Tan, T. K., Sullivan, G., and Wedi, T. (2005). Recommended simulation common conditions for coding efficiency experiments. *ITU-T Q*, 6.

Tanimoto, M., Tehrani, M., Fujii, T., and Yendo, T. (2011). Free-Viewpoint TV. *IEEE Signal Processing Magazine*, 28(1):67 –76.

Tauber, Z., Li, Z.-N., and Drew, M. (2007). Review and Preview: disocclusion by inpainting for image-based rendering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(4):527 –540.

Tian, D., Lai, P.-L., Lopez, P., and Gomila, C. (2009). View synthesis techniques for 3d video. *Proc. SPIE*, 7443:74430T–74430T–11.

Tola, E., Zhang, C., Cai, Q., and Zhang, Z. (2009). Virtual view generation with a hybrid camera array. Technical report.

Tosic, I., Olshausen, B. A., and Culpepper, B. J. (2011). Learning sparse representations of depth. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):941–952.

Tran, L., Pal, C., and Nguyen, T. (2010). View synthesis based on conditional random fields and graph cuts. In *2010 17th IEEE International Conference on Image Processing (ICIP)*, pages 433 –436.

Traver, V. J. and Pla, F. (2003). The Log-polar image representation in pattern recognition tasks. In Perales, F. J., Campilho, A. J. C., Blanca, N. P. d. l., and Sanfeliu, A., editors, *Pattern Recognition and Image Analysis*, number 2652 in Lecture Notes in Computer Science, pages 1032–1040. Springer Berlin Heidelberg.

Tschumperle, D. and Deriche, R. (2005). Vector-valued image regularization with PDEs: a common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506 –517.

Varzi, A. C. and Vieu, L. (2004). *Formal ontology in information systems: Proceedings of the Third International Conference (FOIS-2004)*. IOS Press.

Vetro, A., Yea, S., and Smolic, A. (2008). Toward a 3d video format for auto-stereoscopic displays. volume 7073, pages 70730F–70730F–10.

Vito, C. (2015). *Electronic Imaging & the Visual Arts. EVA 2015 Florence: 13-14 May 2015*. Firenze University Press.

Wang, D., Zhao, Y., Wang, Z., and Chen, H. (2015). Hole-filling for DIBR based on depth and gradient information. *International Journal of Advanced Robotic Systems*, pages 1–6.

Wang, J., An, P., Zuo, Y., You, Z., and Zhang, Z. (2014). High accuracy hole filling for Kinect depth maps. volume 9273, pages 92732L–92732L–17.

Wang, Y.-X. and Zhang, Y.-J. (2011). Image inpainting via weighted sparse non-negative matrix factorization. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 3409 –3412.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612.

Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1):98–117.

Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. SIGGRAPH '00, pages 479–488, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Wexler, Y., Shechtman, E., and Irani, M. (2004). Space-time video completion. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, volume 1, pages I–120 – I–127.

Wexler, Y., Shechtman, E., and Irani, M. (2007). Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476.

Wilmer, A. (2003). Fourier-Mellin based image registration (with GUI) - File Exchange - MATLAB Central.

Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. In *2000 International Conference on Image Processing, 2000. Proceedings*, volume 1, pages 493–496.

Wong, W. K., Choo, C. W., Loo, C. K., and Teh, J. P. (2008). FPGA implementation of log-polar mapping. In *15th International Conference on Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008*, pages 45–50.

Xu, X., Po, L.-M., Cheung, C.-H., Feng, L., Ng, K.-H., and Cheung, K.-W. (2013). Depth-aided exemplar-based hole filling for DIBR view synthesis. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2840–2843.

Xu, X., Po, L. M., Cheung, K. W., Ng, K. H., Wong, K. M., and Ting, C. W. (2012). A foreground biased depth map refinement method for DIBR view synthesis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 805–808.

Xu, Z. and Sun, J. (2010). Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing*, 19(5):1153 –1165.

Yao, C., Tillo, T., Zhao, Y., Xiao, J., Bai, H., and Lin, C. (2014). Depth map driven hole filling algorithm exploiting temporal correlation information. *IEEE Transactions on Broadcasting*, 60(2):394–404.

Zhang, L. and Tam, W. (2005). Stereoscopic image generation based on depth images for 3d TV. *IEEE Transactions on Broadcasting*, 51(2):191 – 199.

Zhu, C. and Li, S. (2016). Depth image based view synthesis: new insights and perspectives on hole generation and filling. *IEEE Transactions on Broadcasting*, 62(1):82–93.

Zhu, C., Zhao, Y., Yu, L., and Tanimoto, M. (2012). *3D-TV system with depth-image-based rendering: architectures, techniques and challenges*. Springer Science & Business Media.

Zinger, S., Do, L., and de With, P. H. N. (2010). Free-viewpoint depth image based rendering. *J. Vis. Comun. Image Represent.*, 21(5-6):533–541.

Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608.

Zokai, S. and Wolberg, G. (2005). Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Transactions on Image Processing*, 14(10):1422–1434.

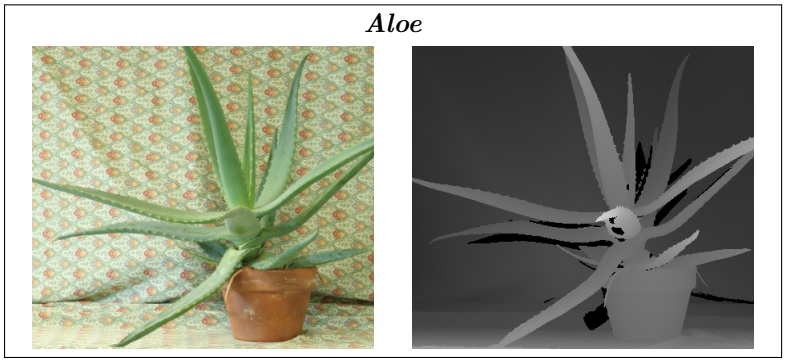
Zone, R. (2007). *Stereoscopic Cinema and the Origins of 3-D Film, 1838-1952*. University Press of Kentucky.

Appendices

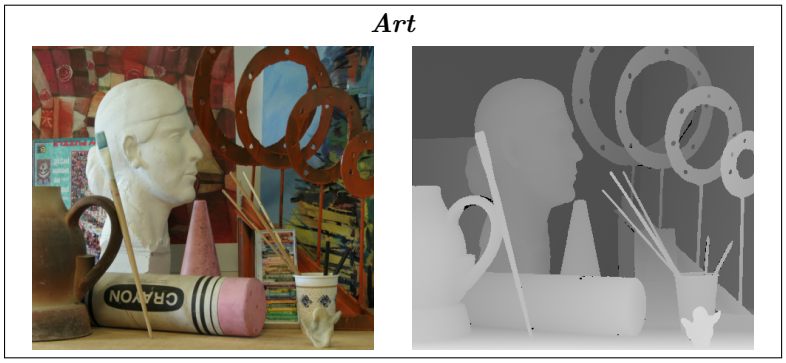
Appendix A

Middlebury Dataset Images

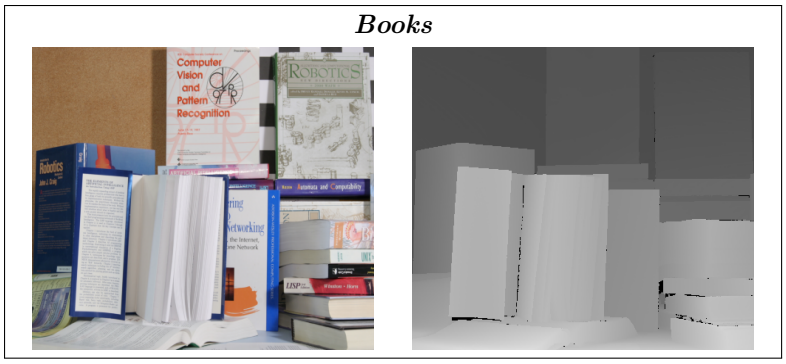
This Appendix includes the images from the Middlebury datasets (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003), used in this thesis. Figure A.1 displays the texture and depth images of *Aloe*, *Art*, *Books* and *Cones*. Similarly, Figure A.2 displays the texture and depth images of *Dolls*, *Laundry, Midd1* and *Teddy*.



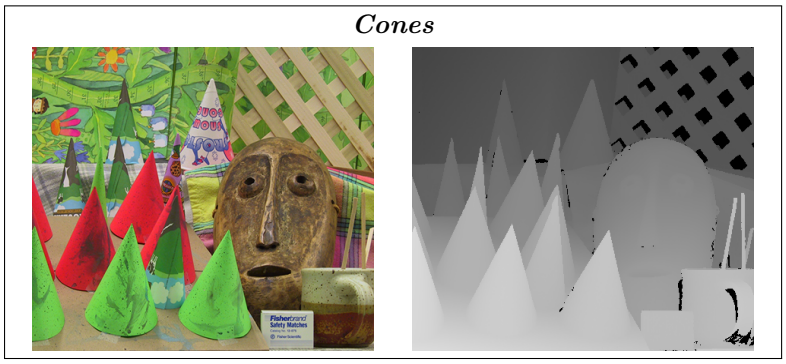
(a)



(b)

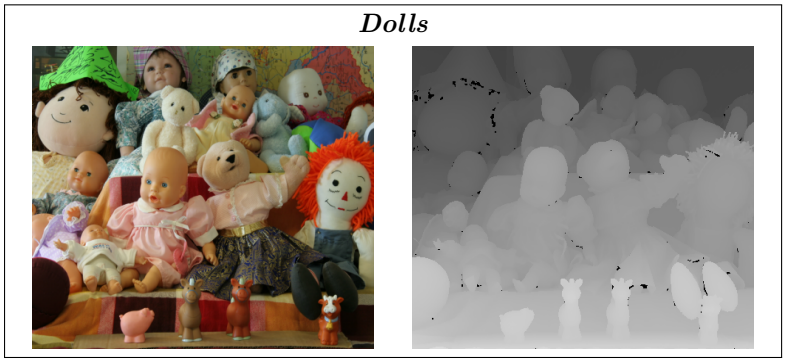


(c)



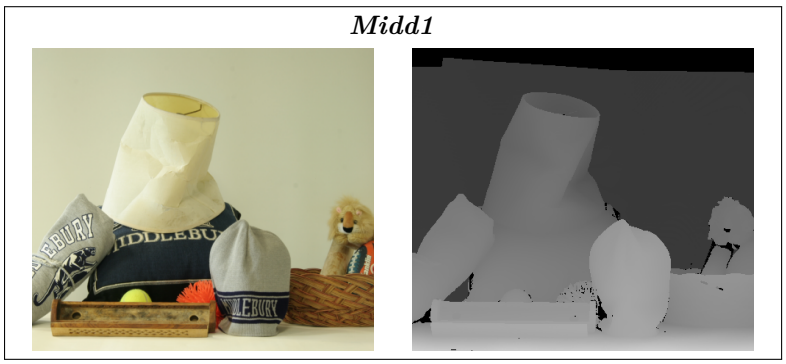
(d)

Figure A.1: Texture and depth images of (a) *Aloe*, (b) *Art*, (c) *Books* and (d) *Cones* from the Middlebury dataset (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003).

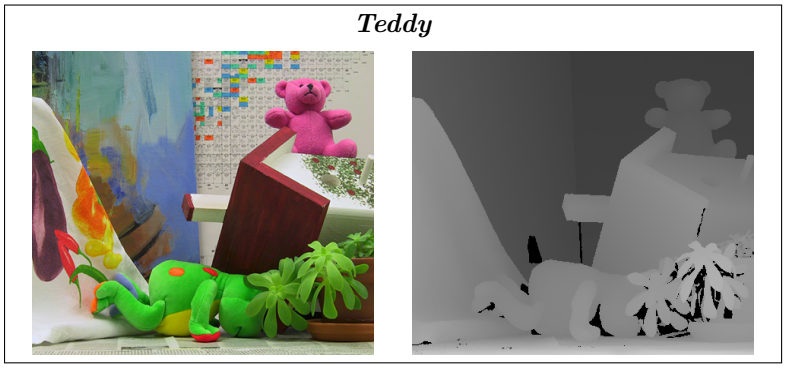


(a)

(b)



(c)



(d)

Figure A.2: Texture and depth images of (a) *Dolls*, (b) *Laundry,* (c) *Midd1* and (d) *Teddy* from the Middlebury dataset (Hirschmuller and Scharstein, 2007; Scharstein and Pal, 2007; Scharstein and Szeliski, 2003).

Appendix B

Supplementary Results for Chapter 4: *Experiment 1* and *2*

This Appendix includes the supplementary results for *Experiment 1* and *2*, discussed in Chapter 4.

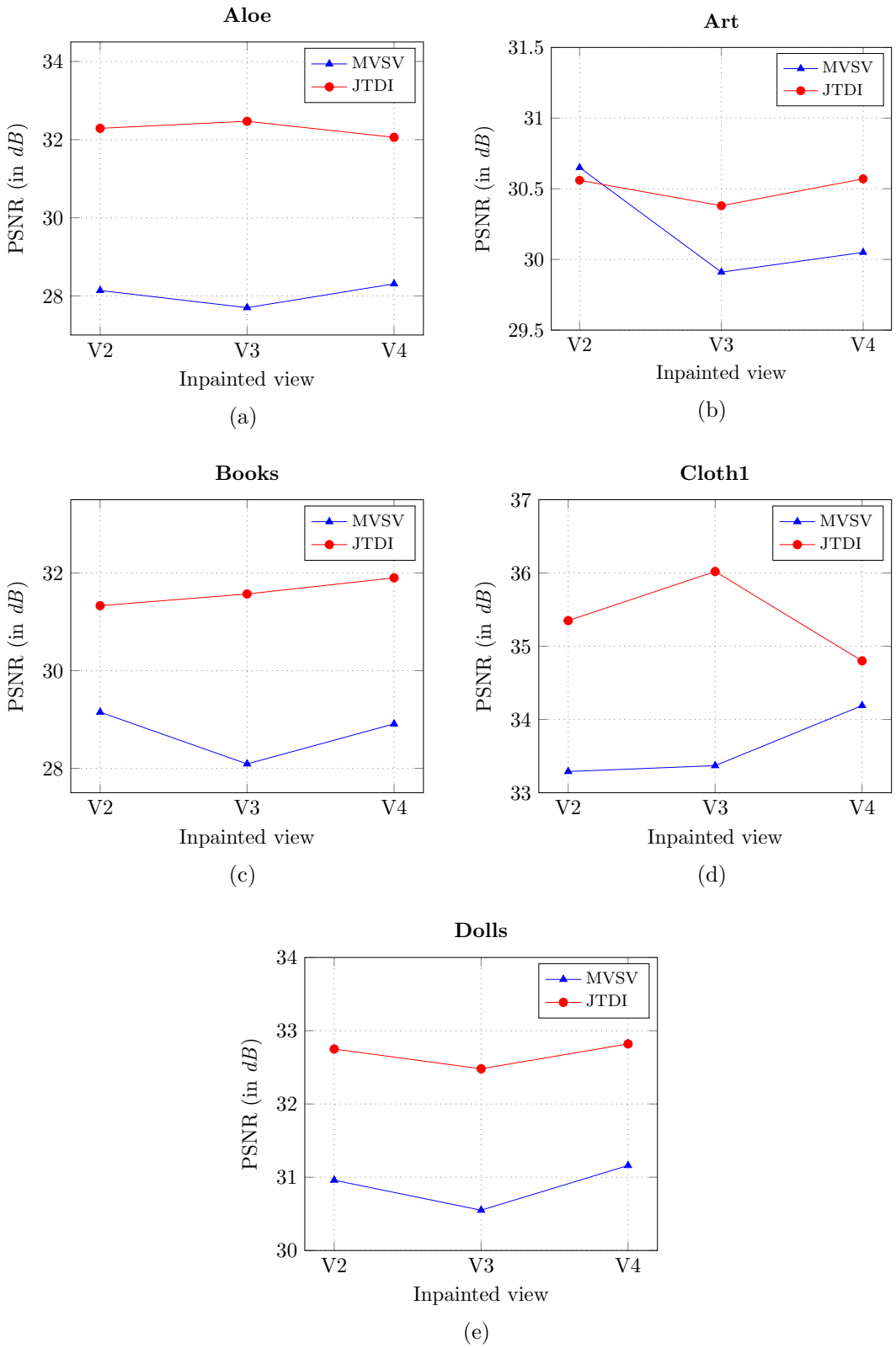


Figure B.1: PSNR results for *Experiment 1*: Inpainting DS-DIBR views. Comparison of three views (V2, V3 and V4) for (a) *Aloe*, (b) *Art*, (c) *Books*, (d) *Cloth1* and (e) *Dolls*, inpainted using MVSV and JTDI.

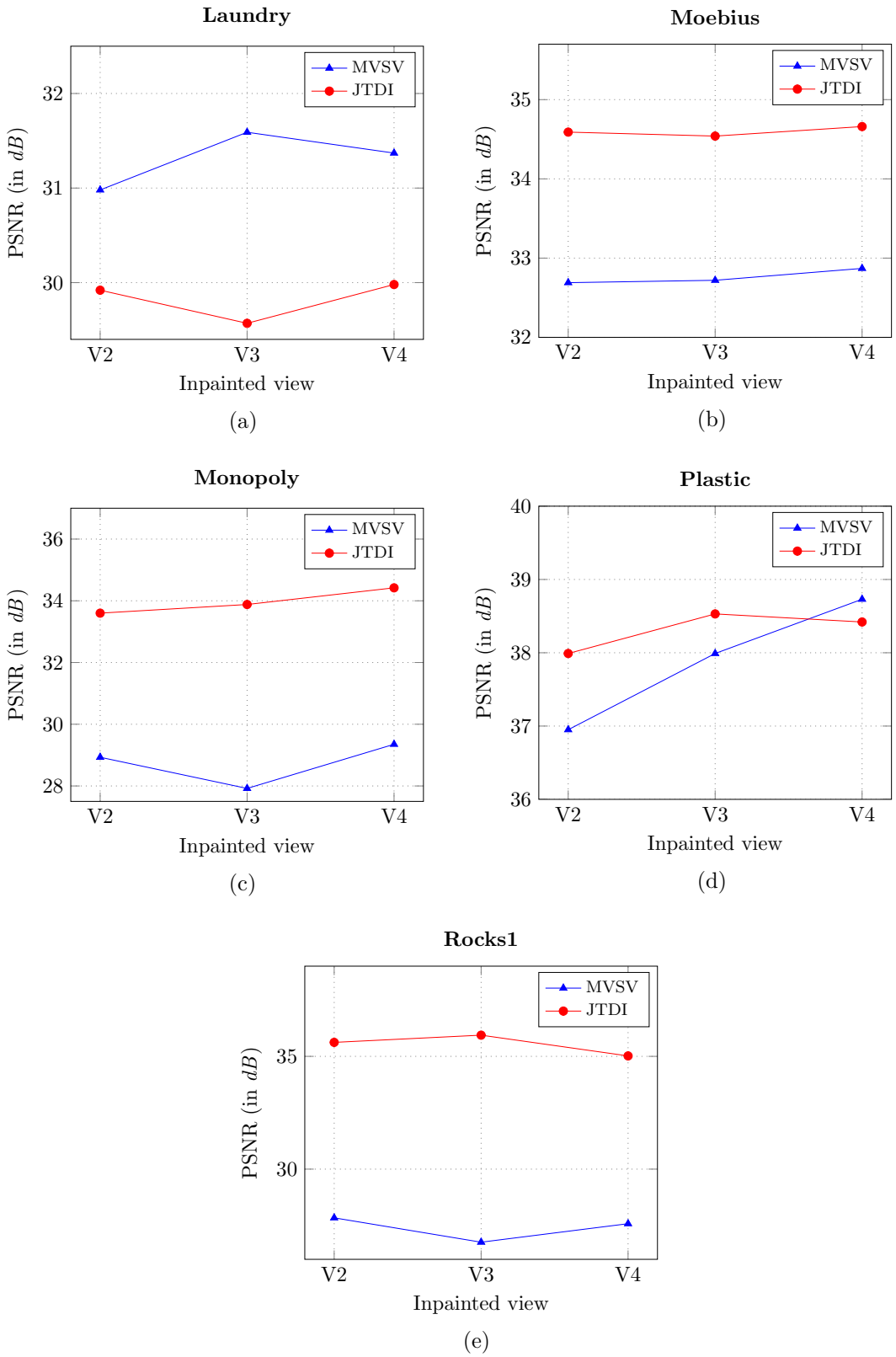


Figure B.2: PSNR results for *Experiment 1*: Inpainting DS-DIBR views. Comparison of three views (V2, V3 and V4) for (a) *Laundry*, (b) *Moebius*, (c) *Monopoly*, (d) *Plastic* and (e) *Rocks1,* inpainted using MVSV and JTDI.

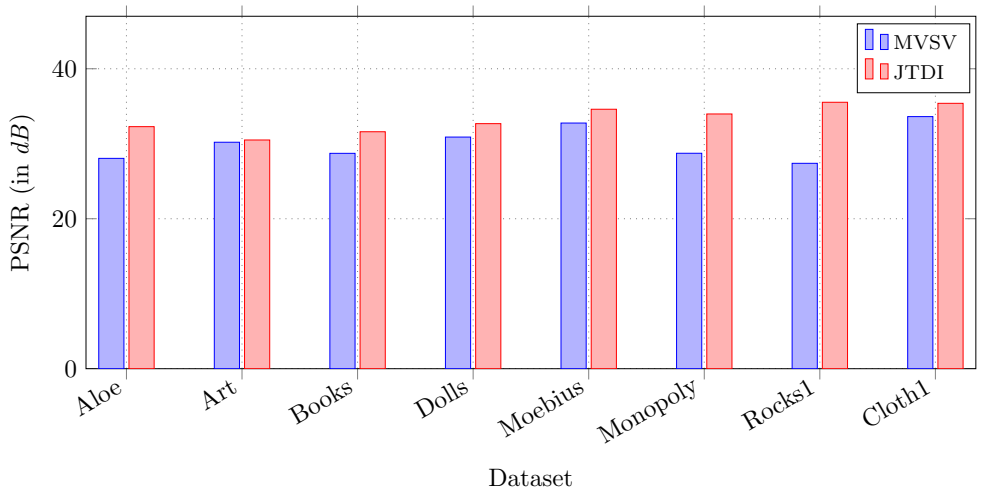


Figure B.3: Average PSNR results for *Experiment 1*: Inpainting DS-DIBR views, using MVSV and JTDI.

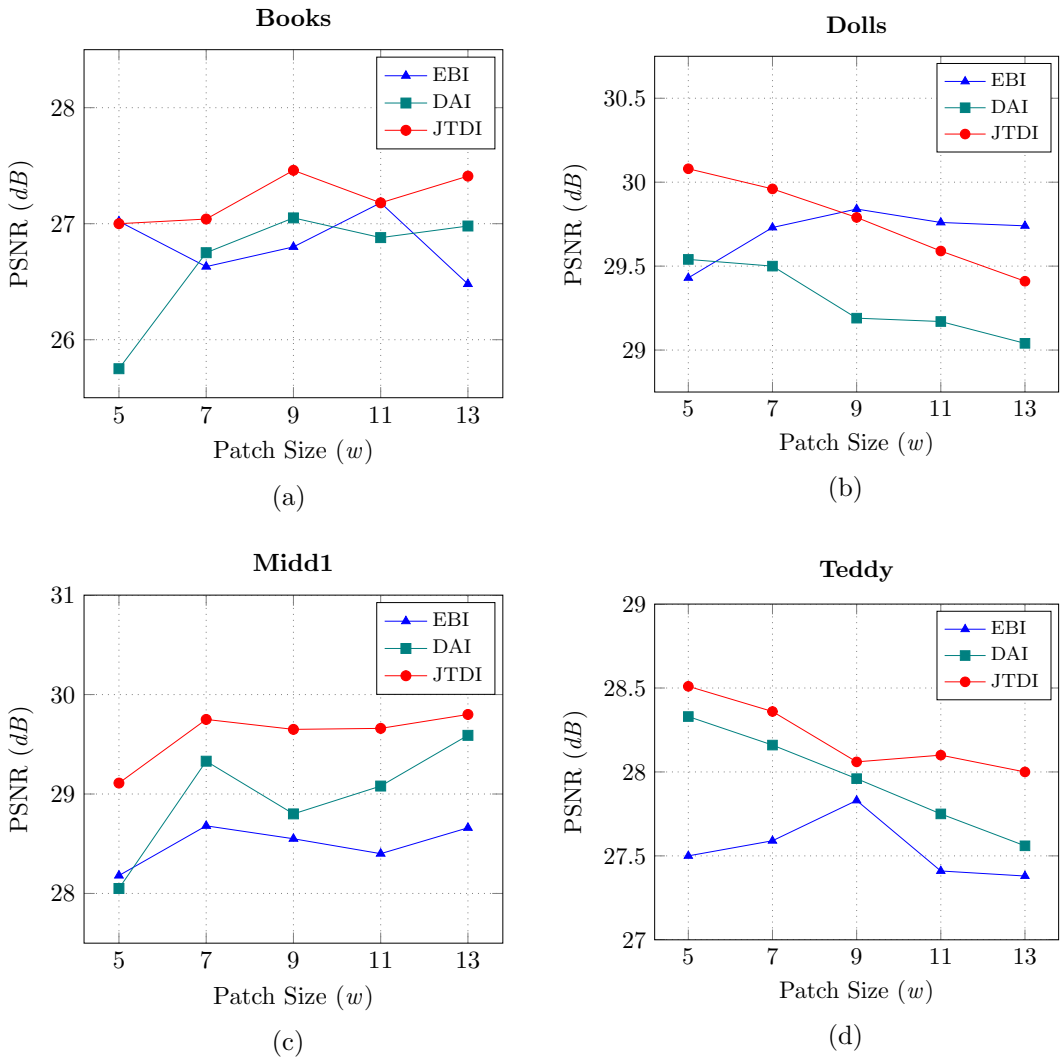


Figure B.4: Whole image PSNR vs patch size results for *Experiment 2*: Inpainting SS-DIBR views, using EBI, DAI and JTDI for (a) *Books* (b) *Dolls* (c) *Midd1* and (d) *Teddy*.

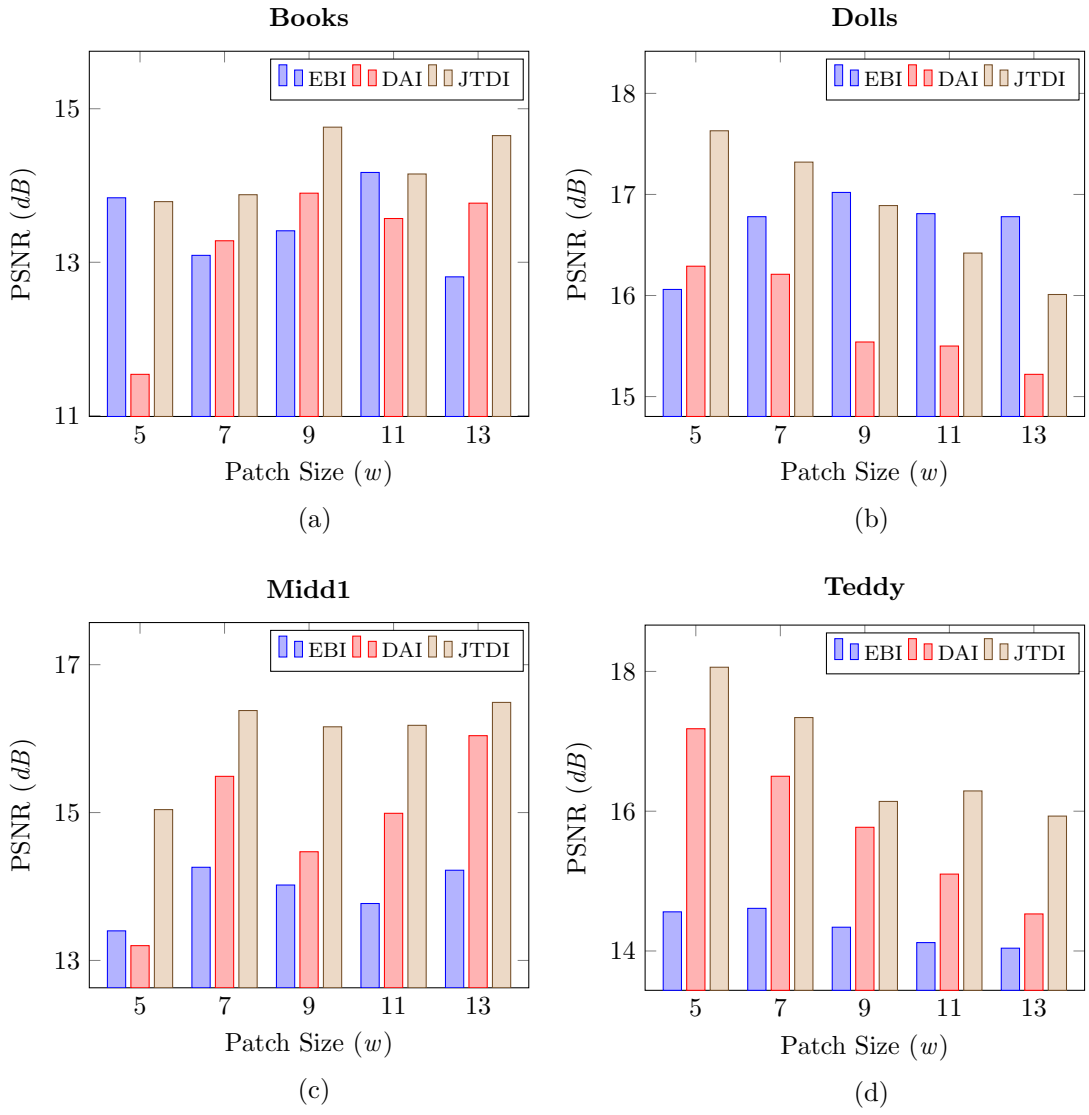


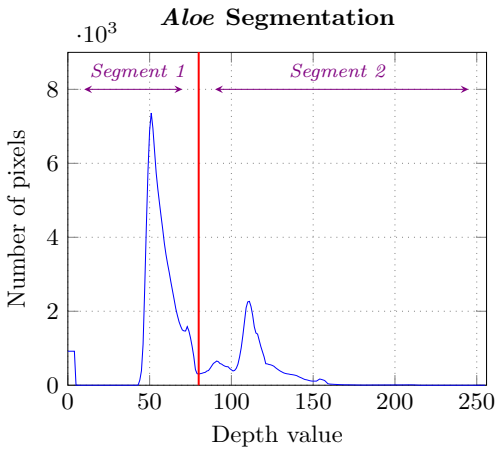
Figure B.5: Inpainted region PSNR vs patch size results for *Experiment 2*: Inpainting SS-DIBR views, using EBI, DAI and JTDI, for (a) *Books* (b) *Dolls* (c) *Midd1* and (d) *Teddy*.

Appendix C

Segmentation Results

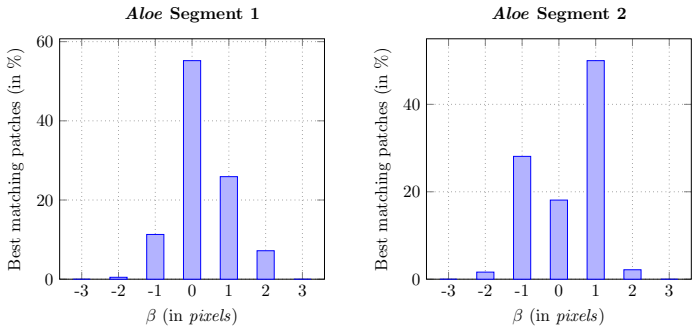
This Appendix includes the results for depth based segmentation of Middlbury datasets. Figures C.1, C.2, C.3 and C.4 show the segmentation of *Aloe*, *Art*, *Books* and *Cones* and their corresponding SP values. Similarly, Figures C.5, C.6, C.7 and C.8 show the segmentation of *Dolls*, *Laundry, Midd1* and *Teddy* and their corresponding SP values.

(a)



(b)

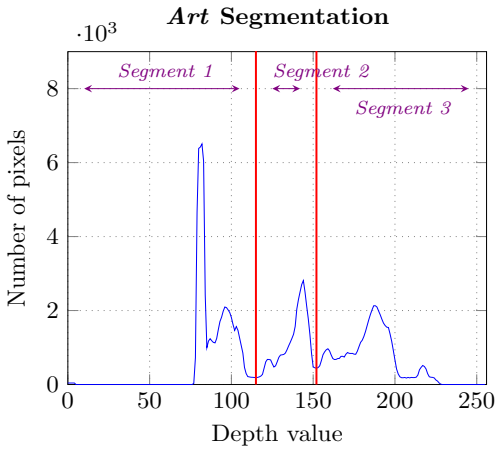
Segment 1 **Segment 2**



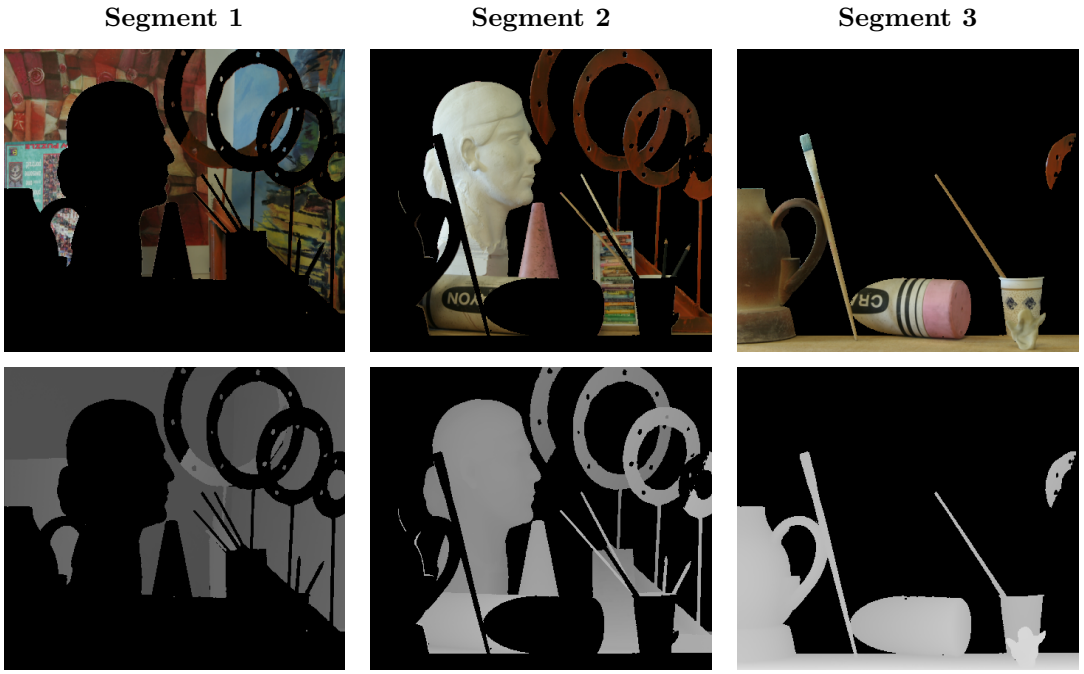
(c) (d)

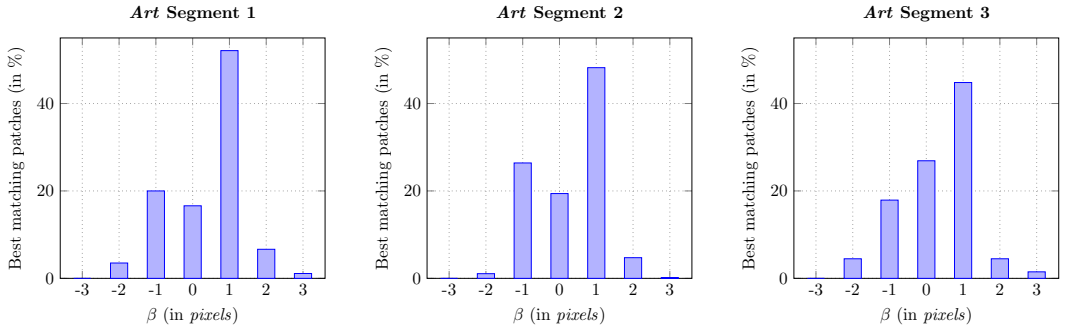
Figure C.1: Segmentation result for *Aloe* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1 and (d) Segment 2, respectively.

(a)



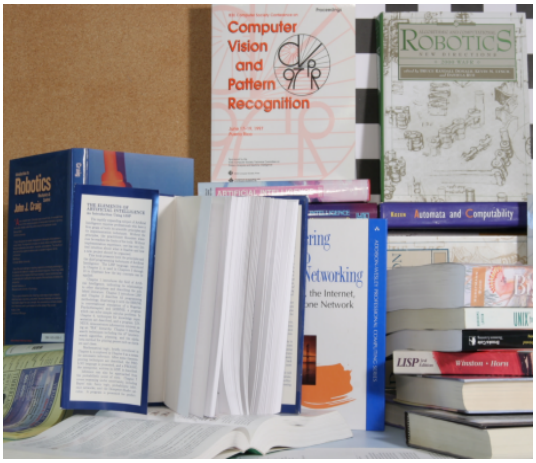
(b)



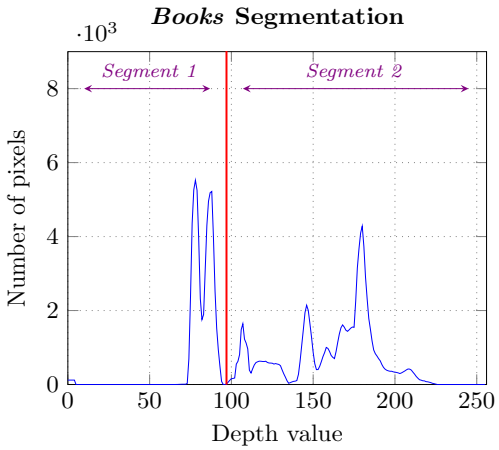


(c) (d) (e)

Figure C.2: Segmentation result for *Art* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1, (d) Segment 2 and (e) Segment 3, respectively.

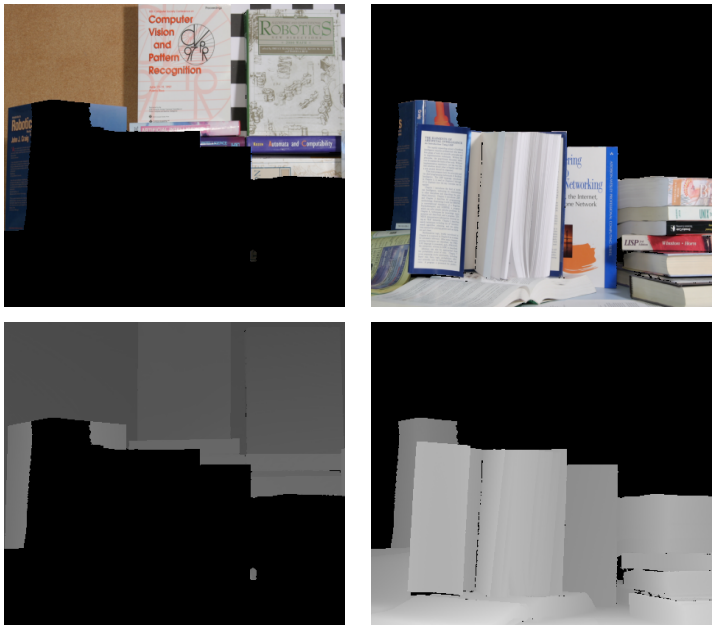


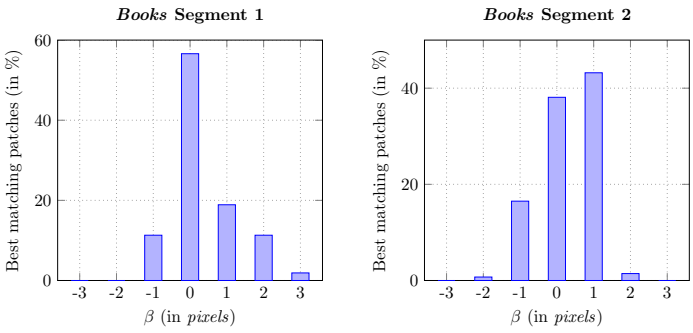
(a)



(b)

Segment 1 **Segment 2**





(c) (d)

Figure C.3: Segmentation result for *Books* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1 and (d) Segment 2, respectively.

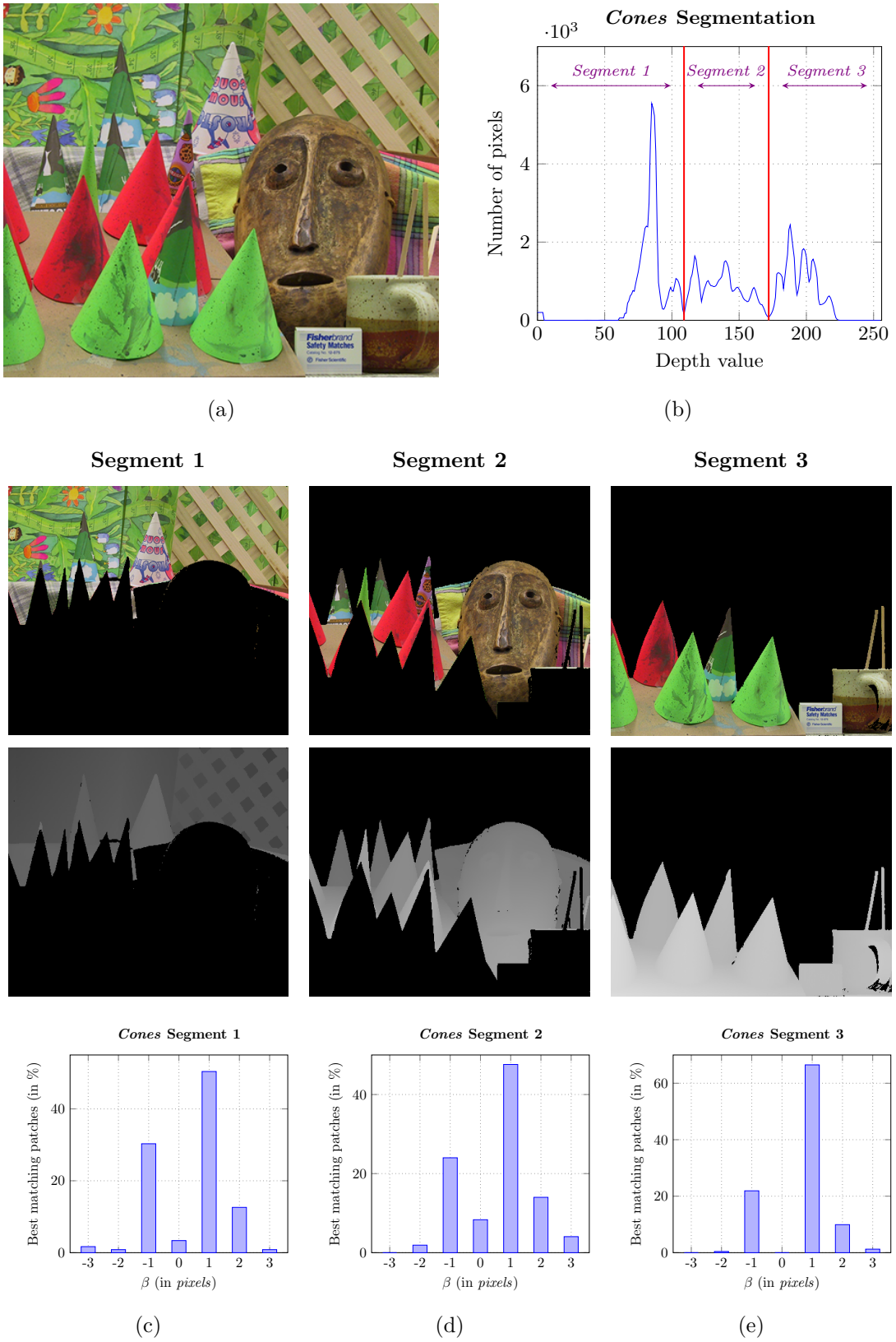


Figure C.4: Segmentation result for *Cones* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1, (d) Segment 2 and (e) Segment 3, respectively.

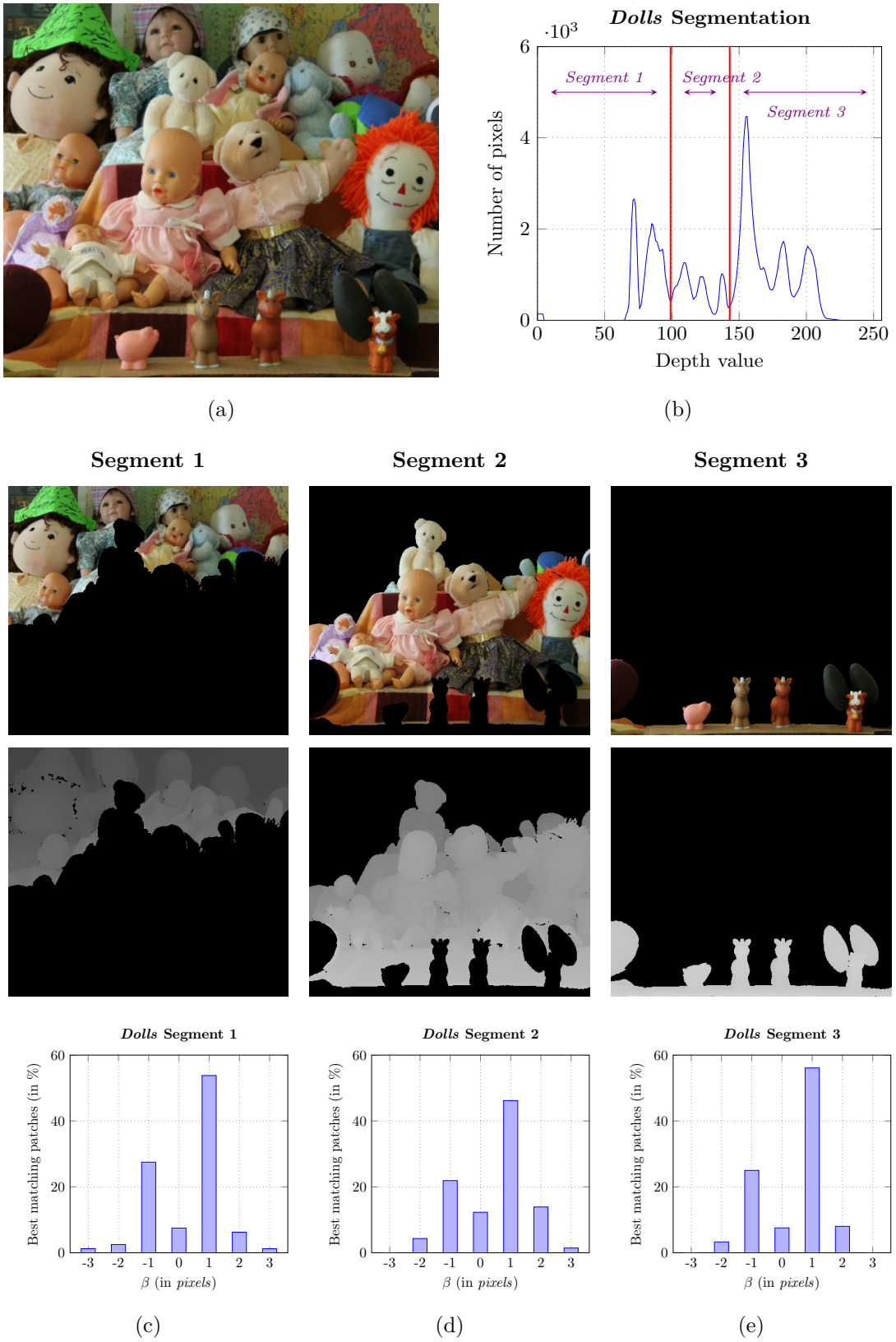


Figure C.5: Segmentation result for *Dolls* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1, (d) Segment 2 and (e) Segment 3, respectively.

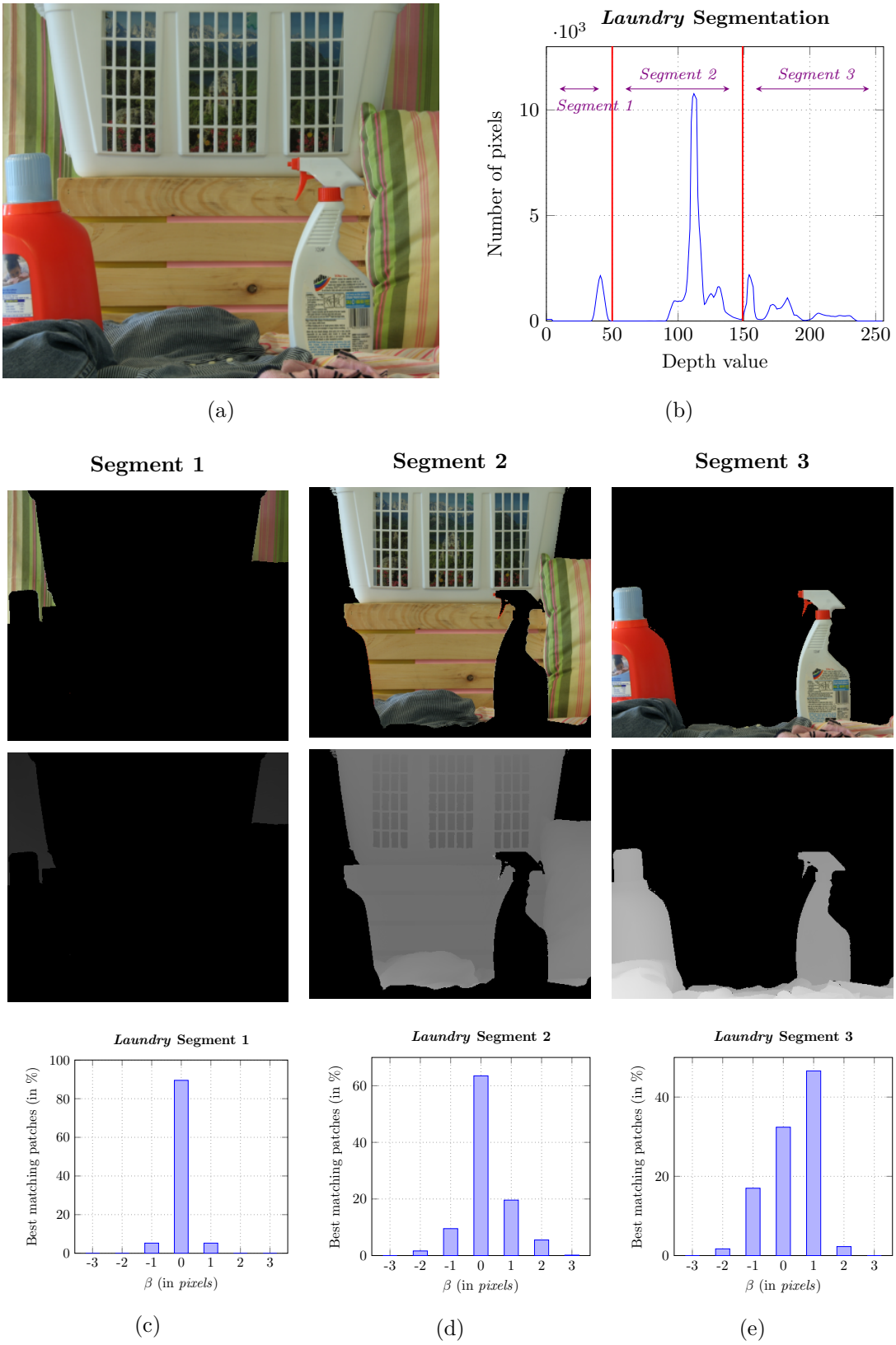


Figure C.6: Segmentation result for *Laundry* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1, (d) Segment 2 and (e) Segment 3, respectively.

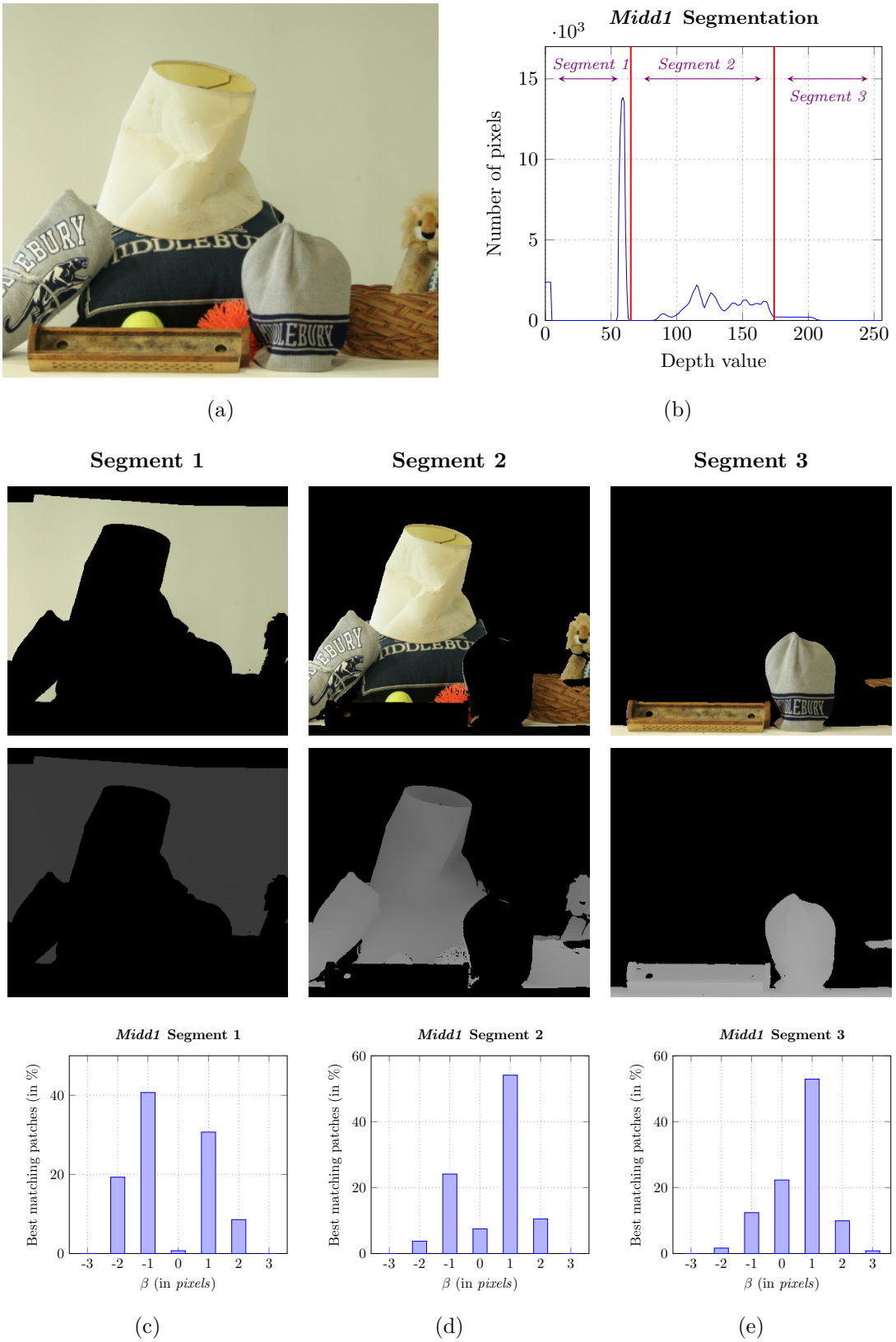
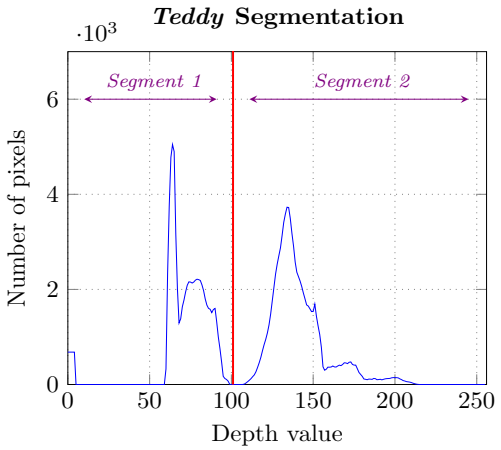


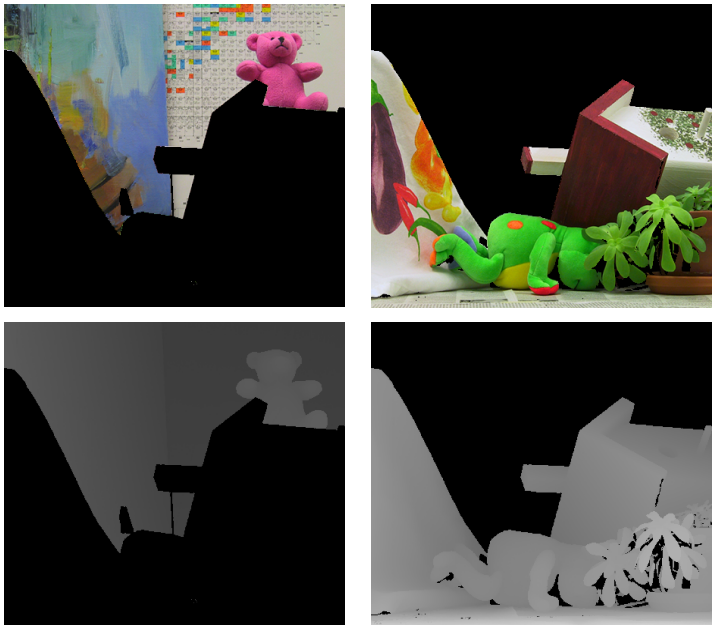
Figure C.7: Segmentation result for *Midd1* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1, (d) Segment 2 and (e) Segment 3, respectively.

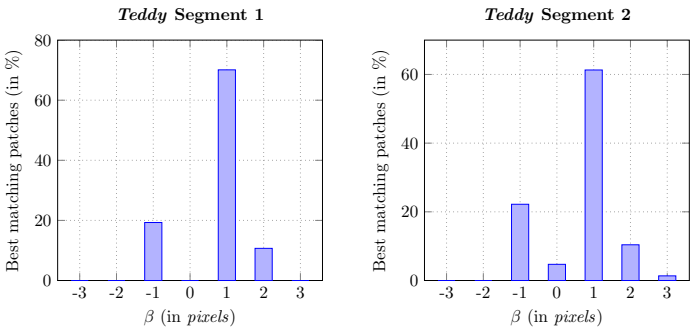
(a)



(b)

Segment 1 **Segment 2**





(c) (d)

Figure C.8: Segmentation result for *Teddy* (a) whole image, (b) depth histogram, and texture, depth images and corresponding SP values of (c) Segment 1 and (d) Segment 2, respectively.

Appendix D

Supplementary Quantitative Results for Chapter 4, 5 and 6

This Appendix includes supplementary quantitative results for Chapter 4, 5 and 6.

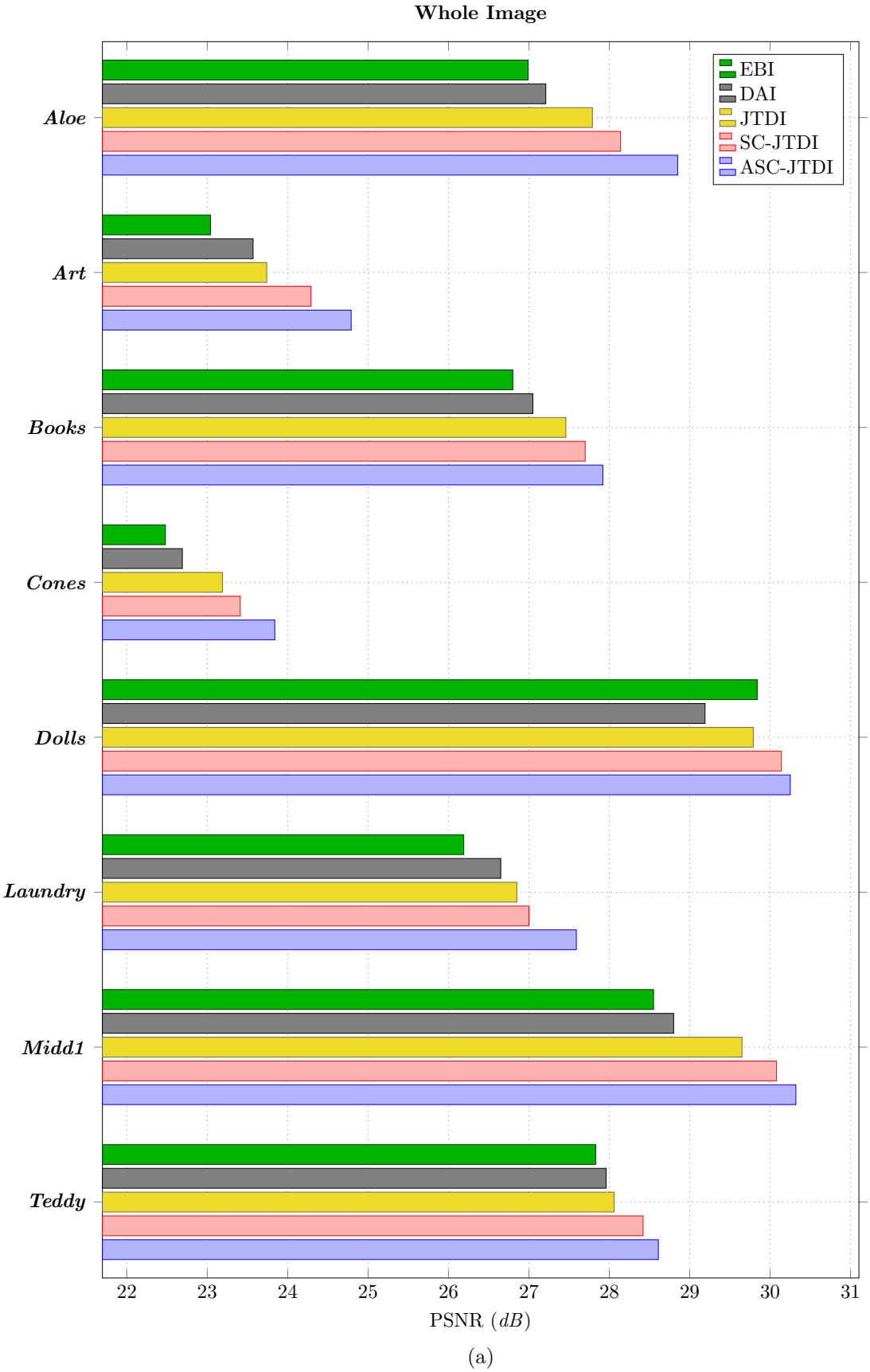


Figure D.1: Whole image PSNR comparison for various image datasets.

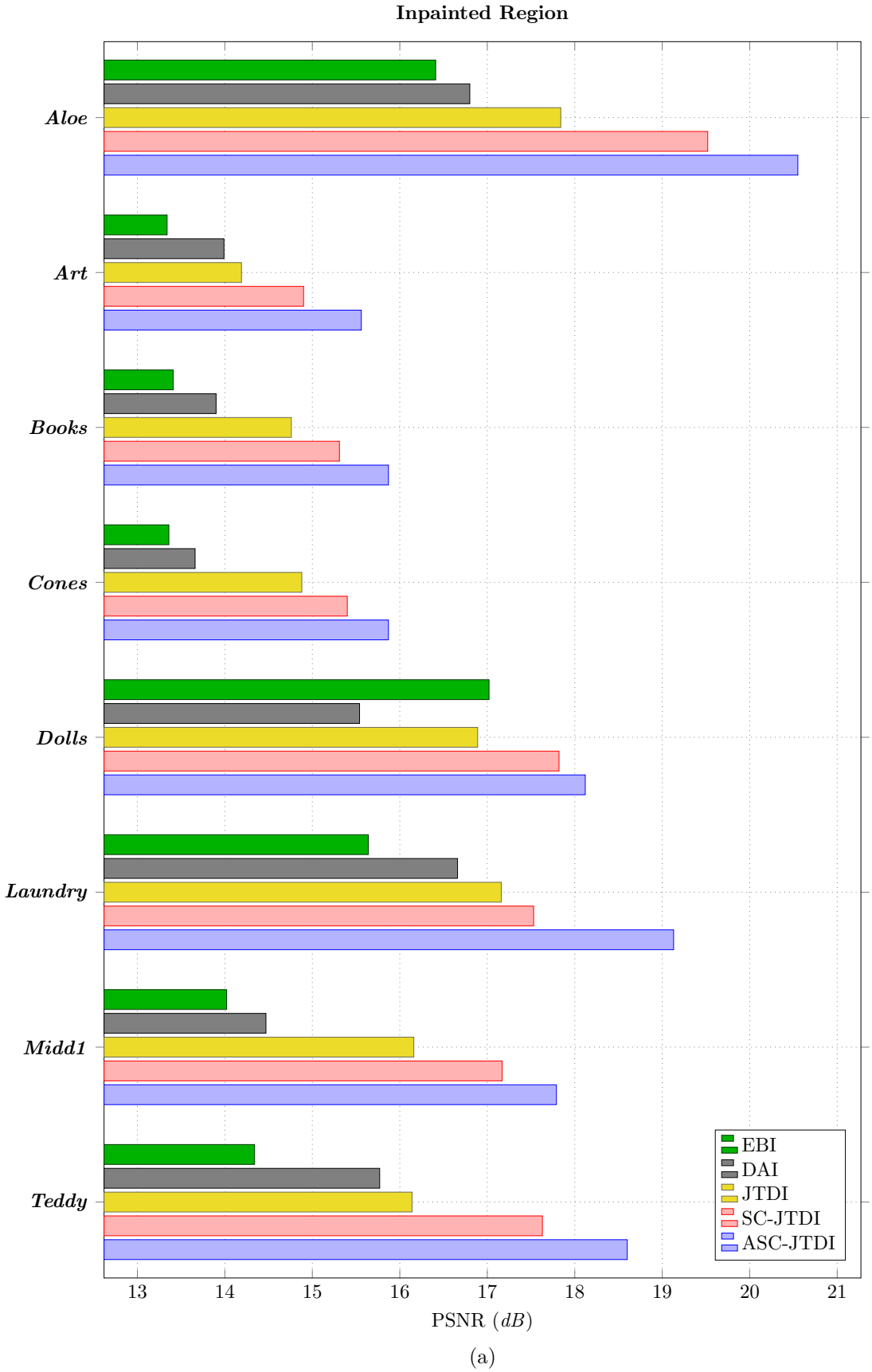


Figure D.2: Inpainted region PSNR comparison for various image datasets.

Dataset	**Segment**	**SP**	**SR**
Aloe	1	{0, 1}	{(1, 0), (1.1, 0), (1, 1), (1, -2)}
	2	{-1, 1}	{(1, 0)}
Art	1	{-1, 1}	{(1, 0), (1.1, 0), (1, -1)}
	2	{-1, 1}	{(1, 0)}
	3	{0, 1}	{(1, 0)}
Books	1	{0}	{(1, 0), (1.1, 1)}
	2	{0, 1}	{(1, 0), (1.1, -2), (1, -1)}
Cones	1	{-1, 1}	{(1, 0), (1.1, 1), (1.2, -1)}
	2	{-1, 1}	{(1, 0), (1.2, 2)}
	3	{-1, 1}	{(1, 0)}
Dolls	1	{-1, 1}	{(0.9, -1), (1.1, 0)}
	2	{-1, 1}	{(1, -1),(1.1, 0)}
	3	{-1, 1}	{(1, 0)}
Laundry	1	{0}	{(1, 0), (1.1, 1)}
	2	{0, 1}	{(1, 0), (0.9, 0)}
	3	{0, 1}	{(1, 0)}
Midd1	1	{-1, 1}	{*none*}
	2	{-1, 1}	{(1, 2)}
	3	{0, 1}	(1, 0)
Teddy	1	{-1, 1}	{(1.1, 2), (1, -1), (1, 2)}
	2	{-1, 1}	{1, 1}

Table D.1: Chosen SP and SR parameters for various image datasets in Chapter 5 and 6 respectively.

Appendix E

Supplementary Qualitative Results for Chapter 4, 5 and 6

This Appendix includes supplementary qualitataive results for Chapter 4, 5 and 6.

(a)

(b)

Figure E.1: *Aloe* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.2: *Aloe* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.3: *Aloe* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.4: *Aloe* texture image inpainted using ASC-JTDI.

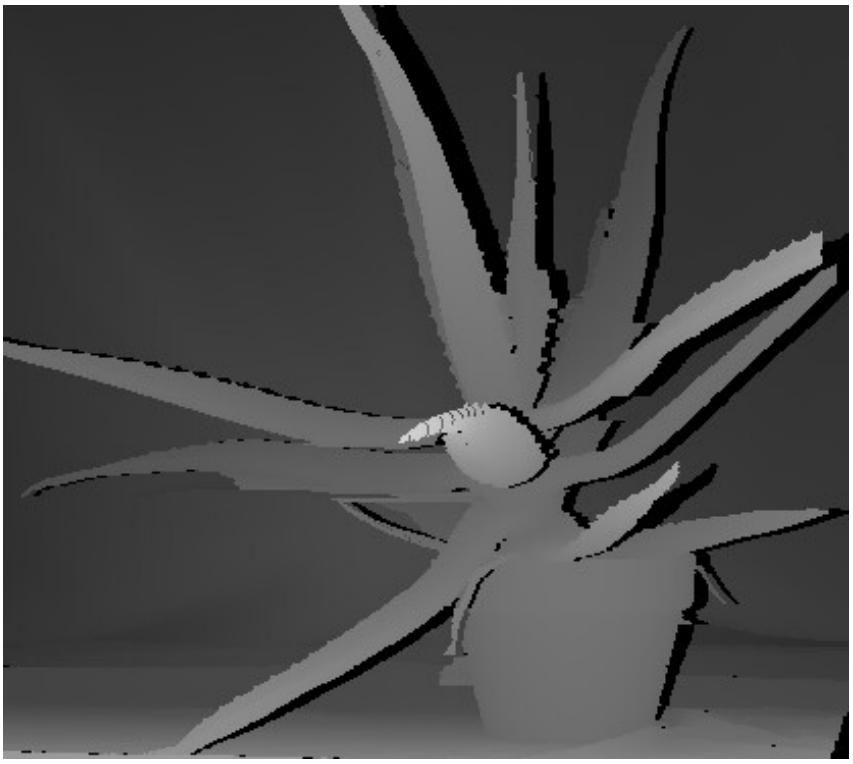
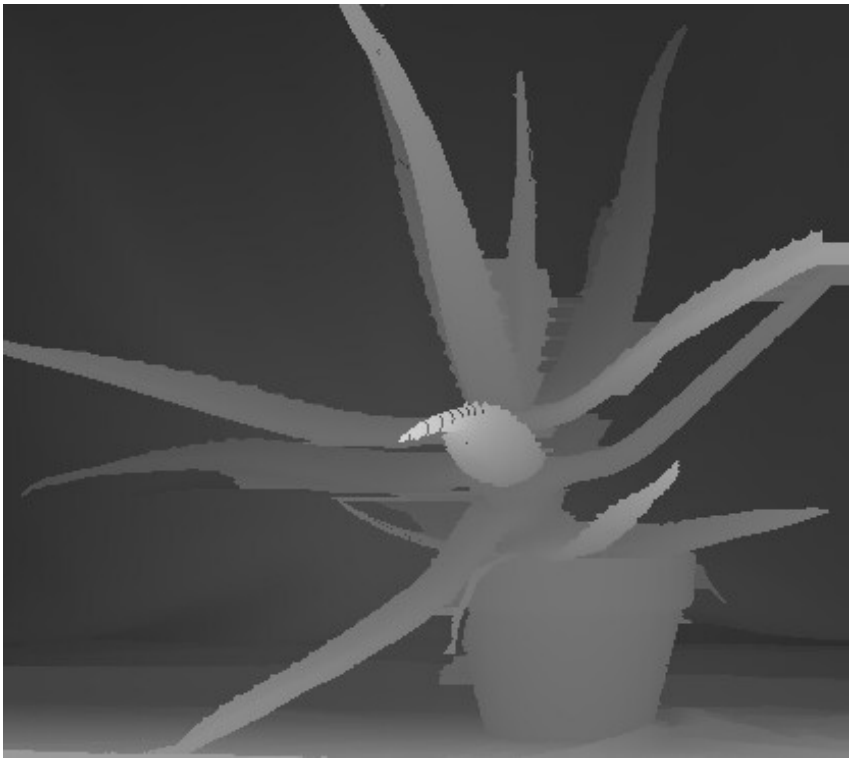
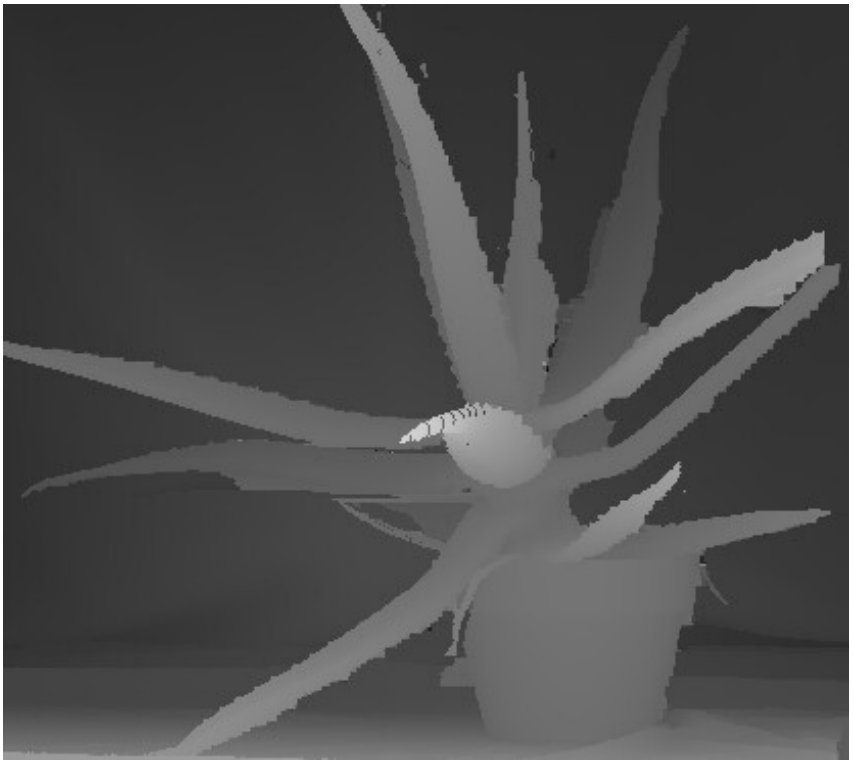


Figure E.5: *Aloe* depth image with holes.



(a)



(b)

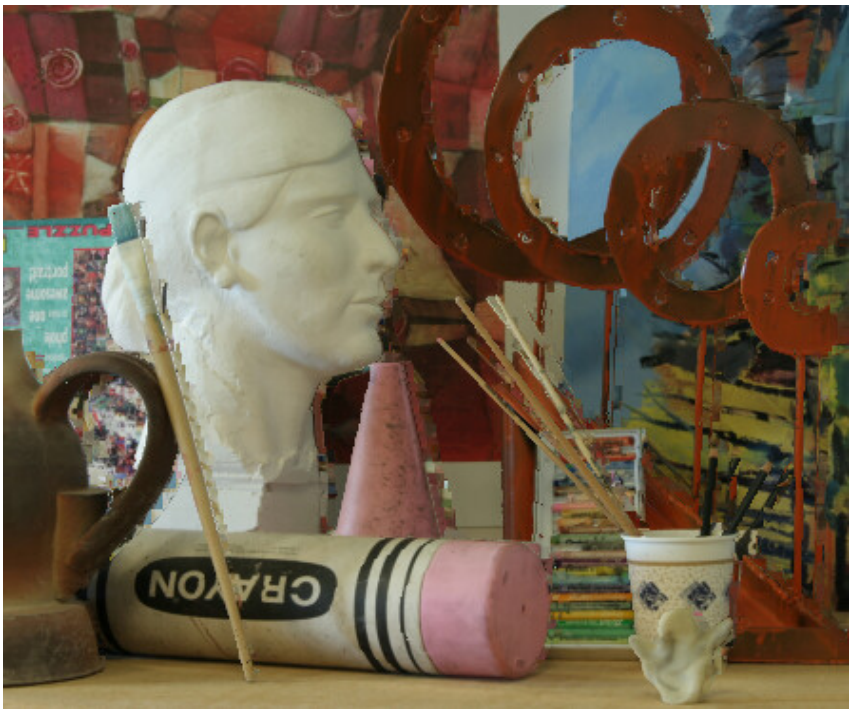
Figure E.6: *Aloe* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.7: *Art* texture image with (a) holes and its corresponding (b) ground truth.

(a)



(b)

Figure E.8: *Art* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.9: *Art* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.10: *Art* texture image inpainted using ASC-JTDI.

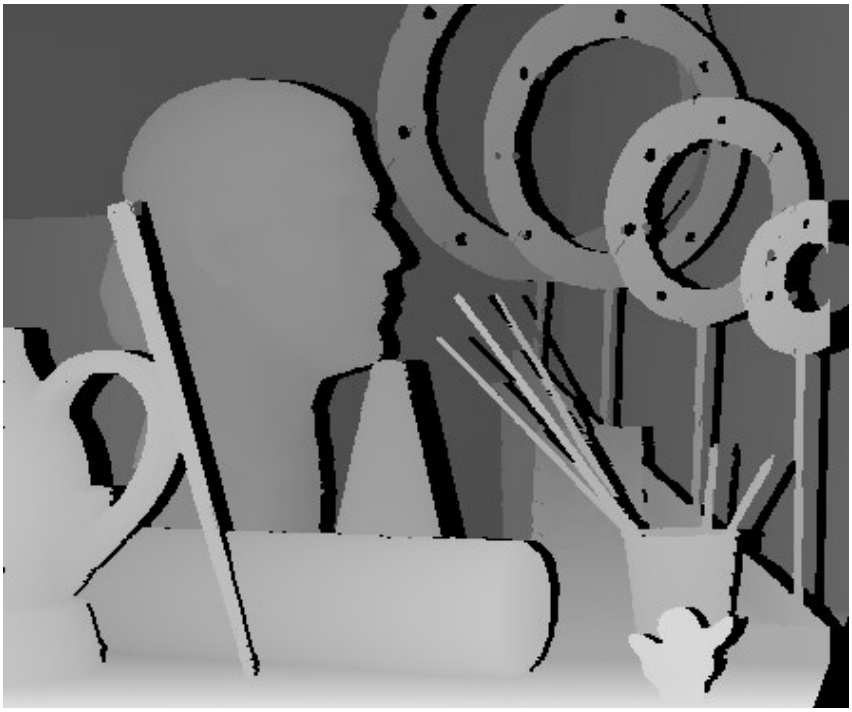
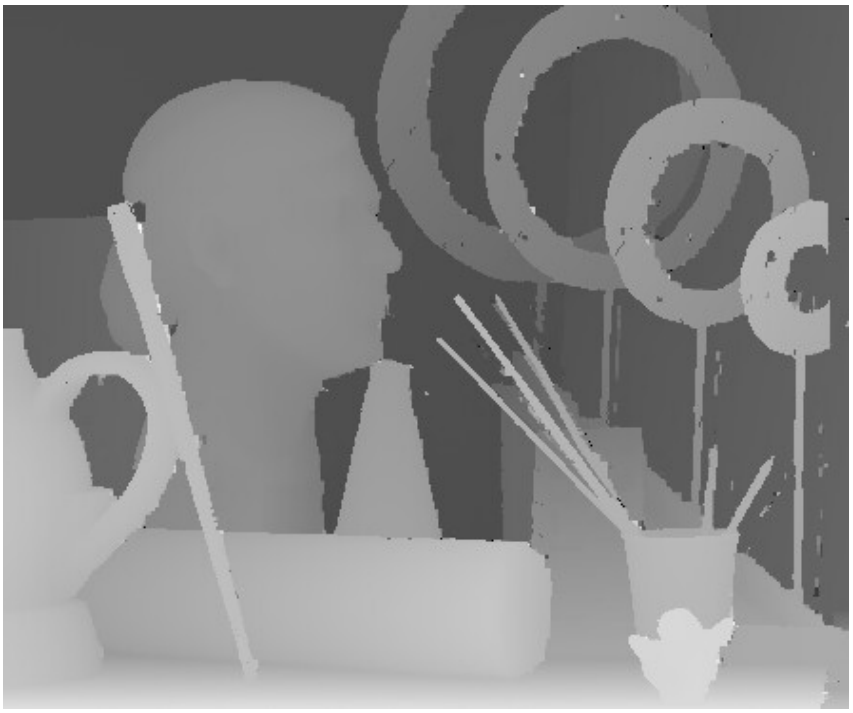


Figure E.11: *Art* depth image with holes.

(a)



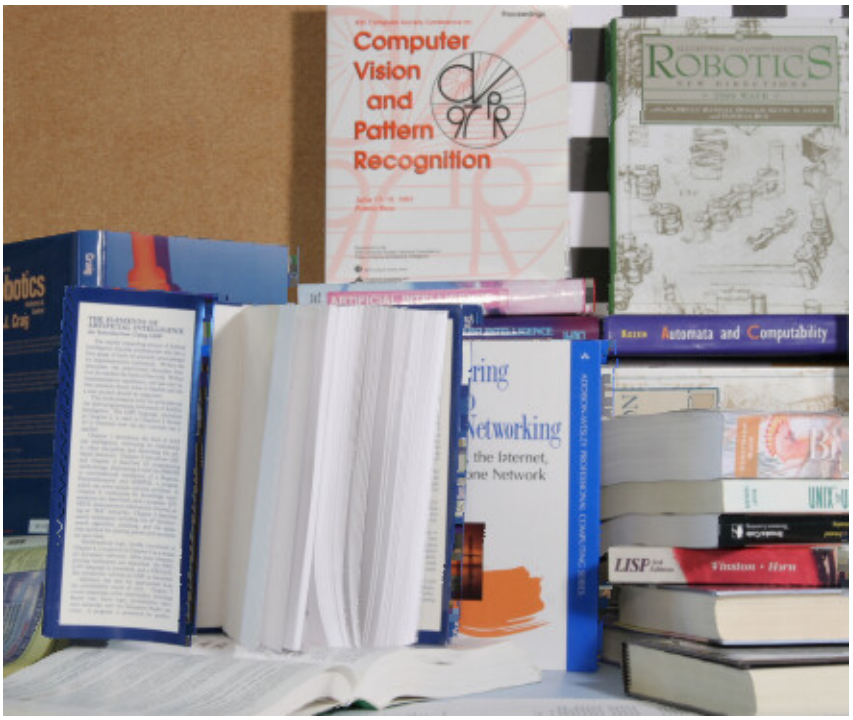
(b)

Figure E.12: *Art* depth image inpainted using (a) Extrapolation and (b) JTDI.

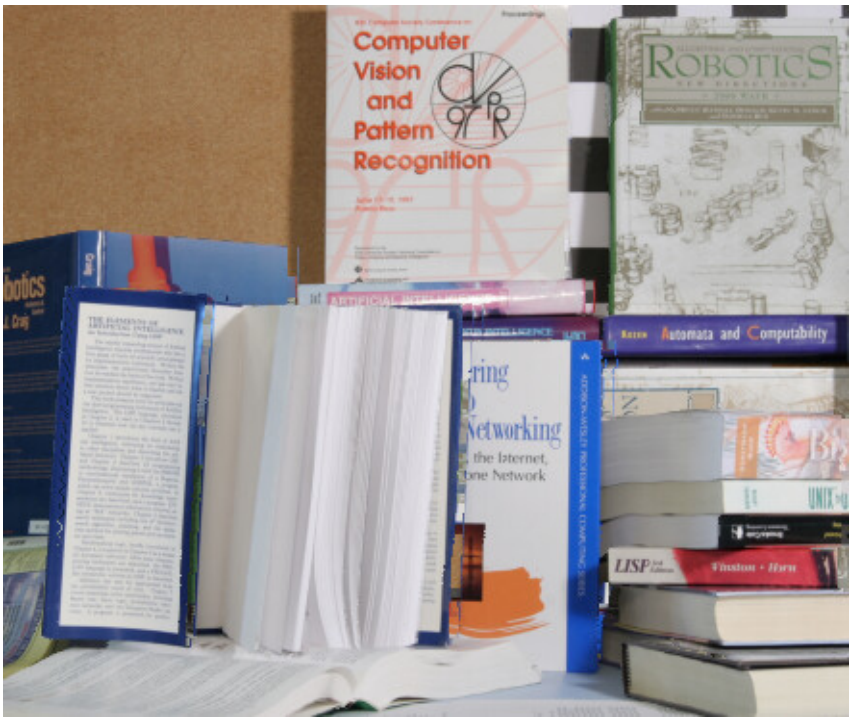
(a)

(b)

Figure E.13: *Books* texture image with (a) holes and its corresponding (b) ground truth.

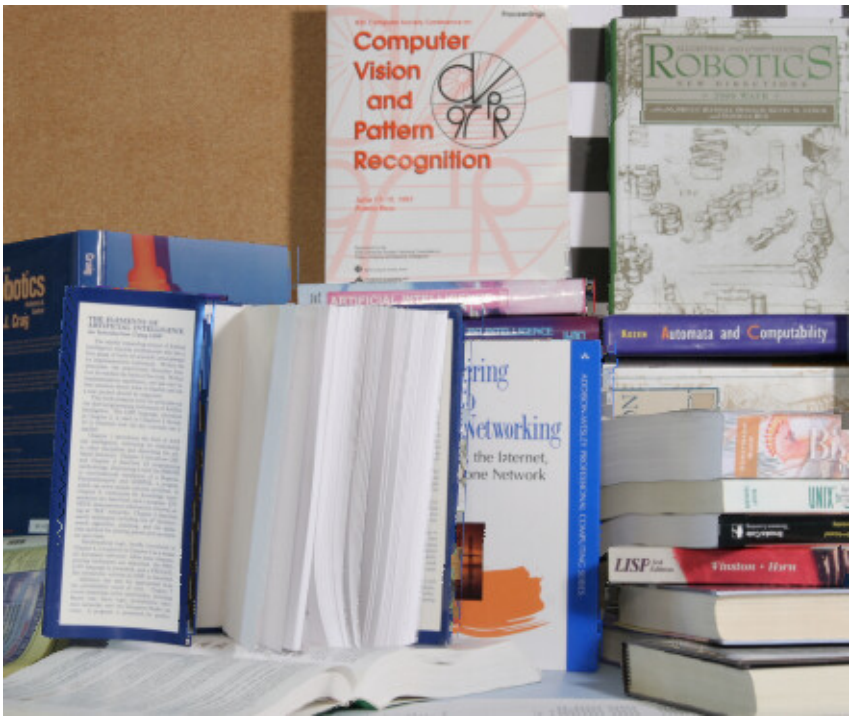


(a)



(b)

Figure E.14: *Books* texture image inpainted using (a) EBI and (b) DAI.



(a)

(b)

Figure E.15: *Books* texture image inpainted using (a) JTDI and (b) SC-JTDI.

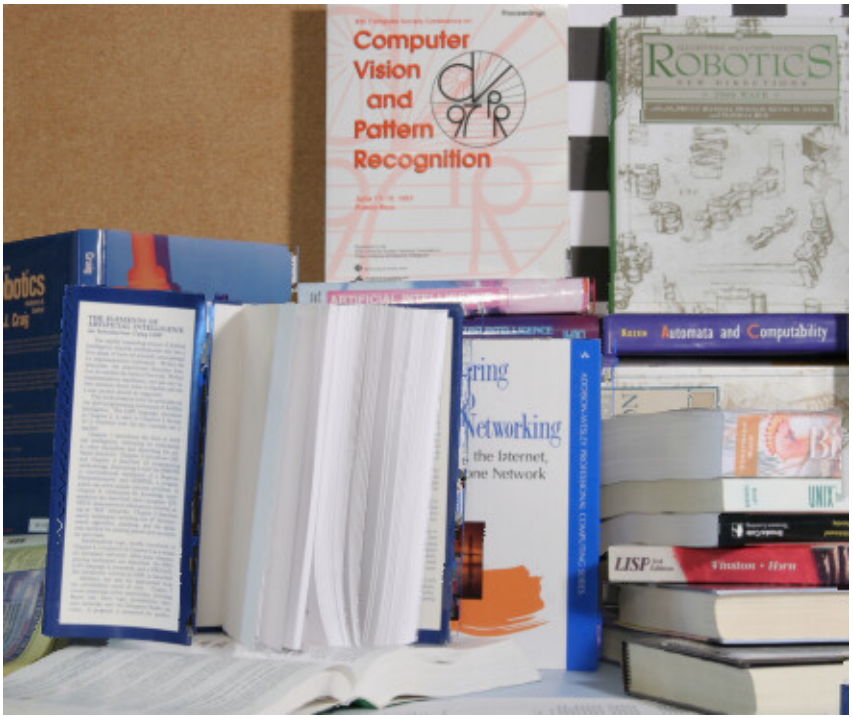


Figure E.16: *Books* texture image inpainted using ASC-JTDI.

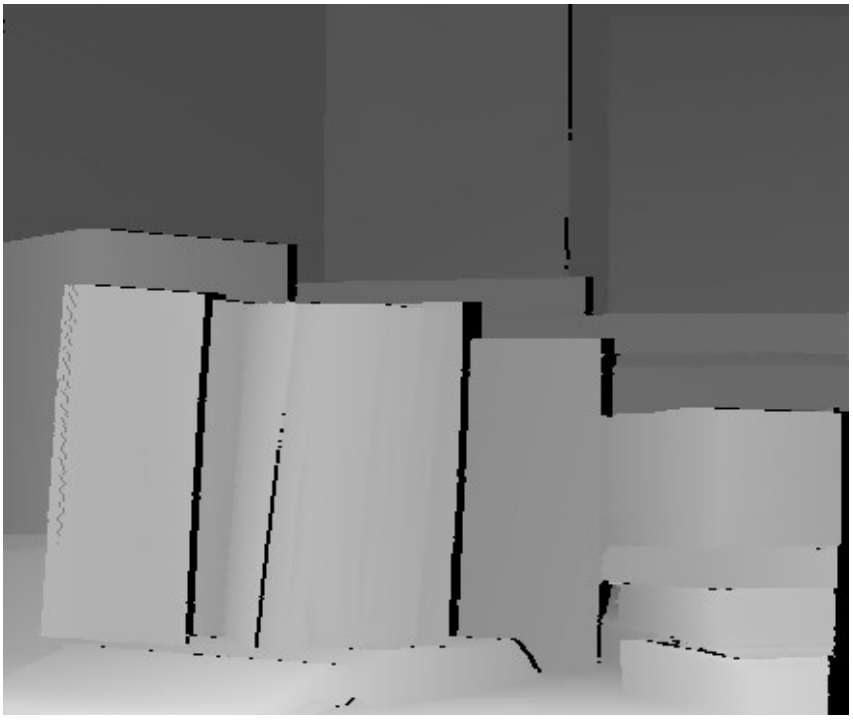
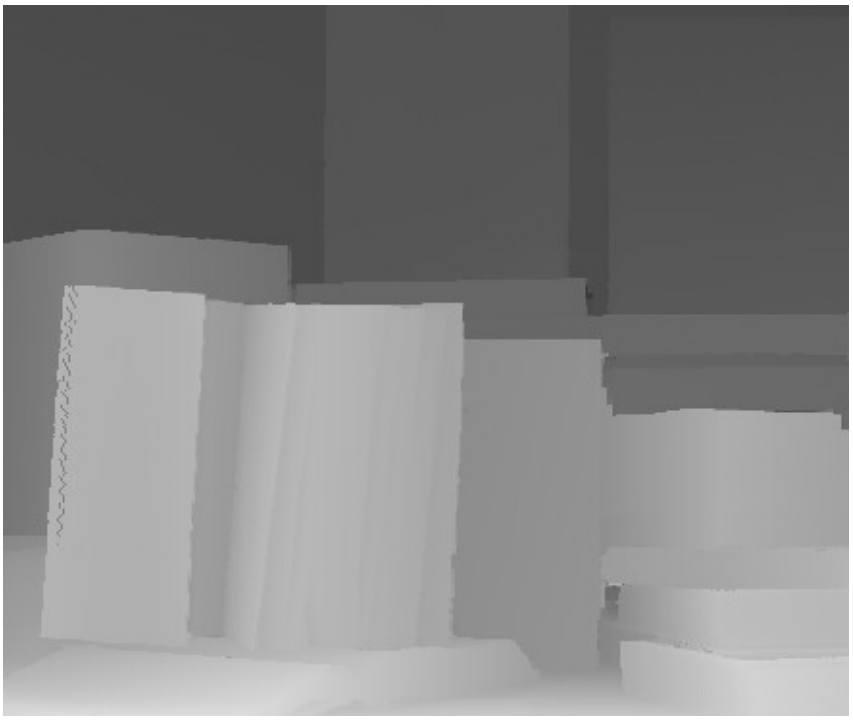
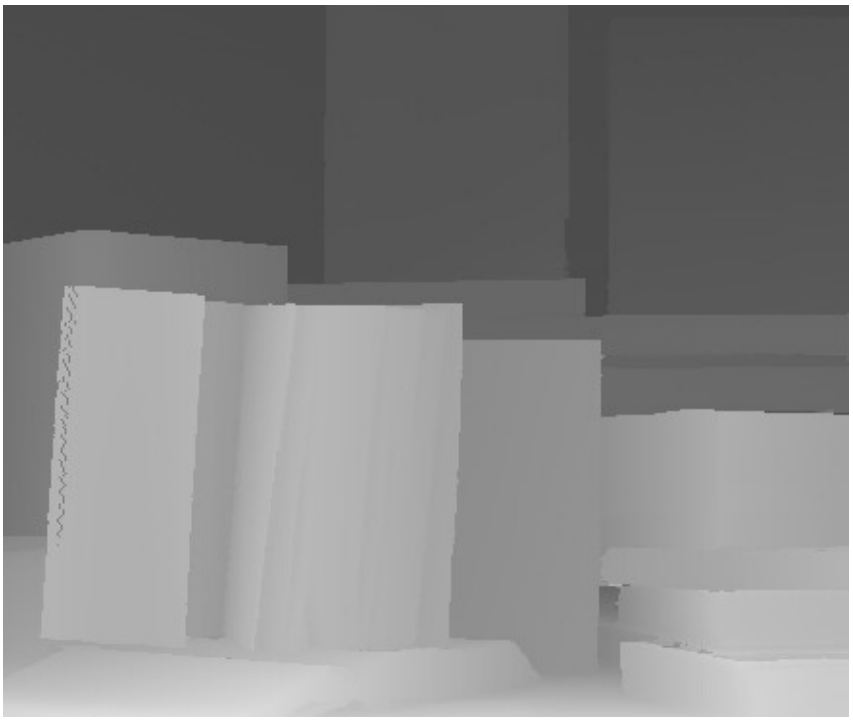


Figure E.17: *Books* depth image with holes.



(a)



(b)

Figure E.18: *Books* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.19: *Cones* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.20: *Cones* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.21: *Cones* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.22: *Cones* texture image inpainted using ASC-JTDI.

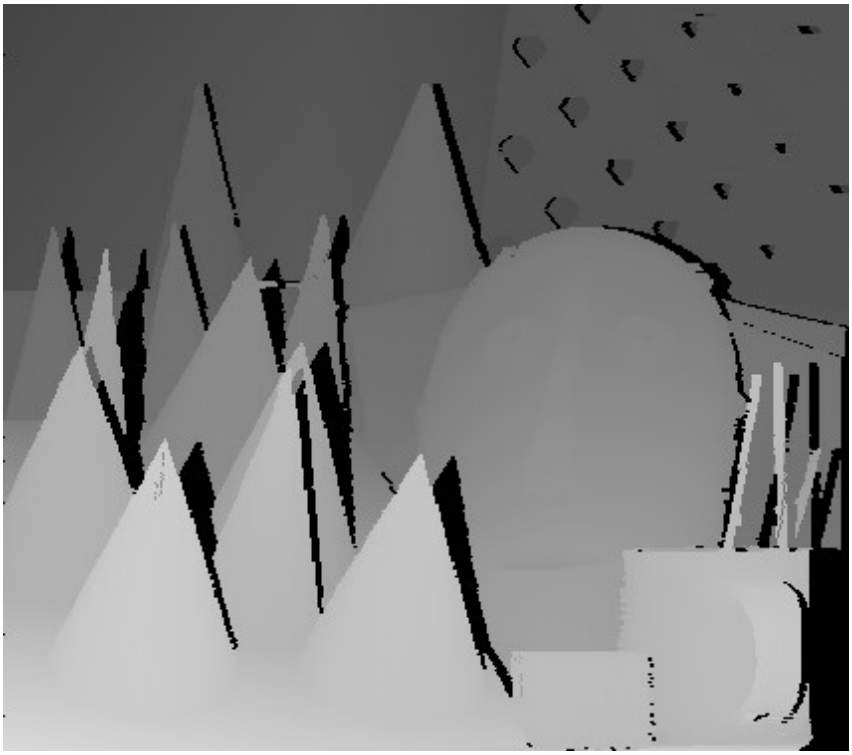
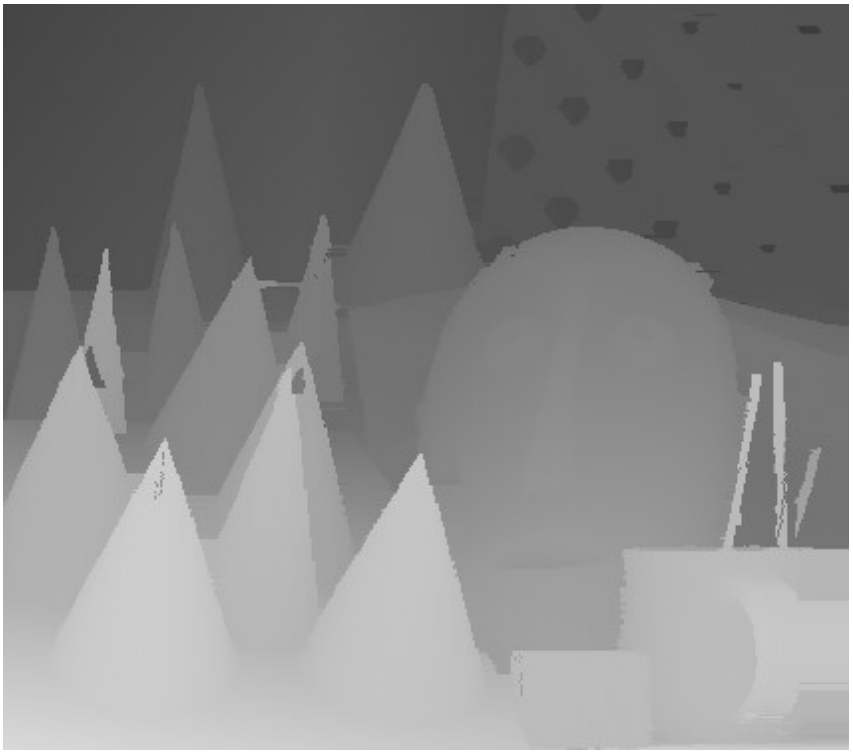
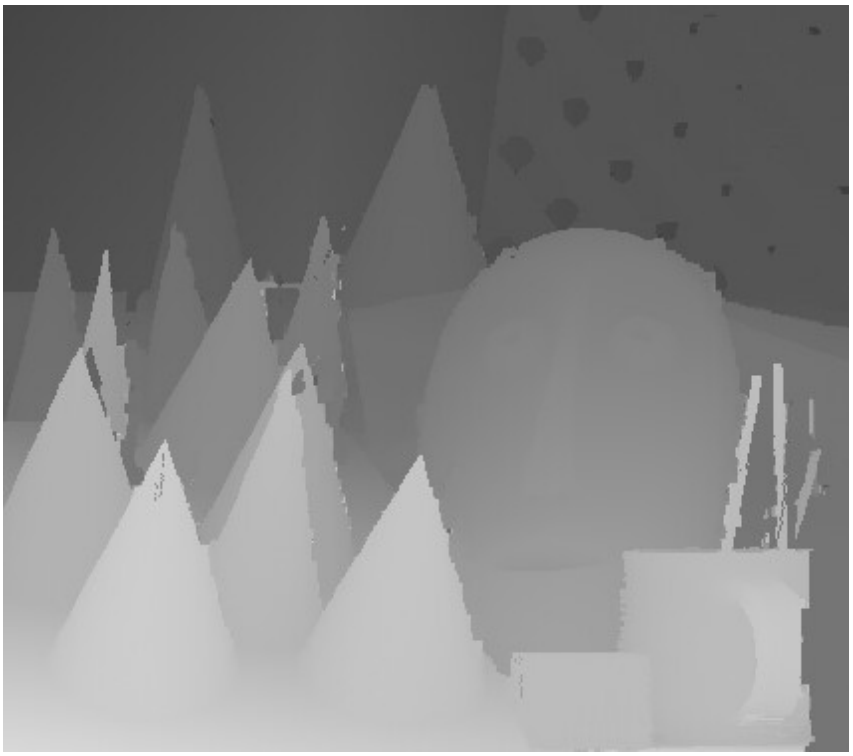


Figure E.23: *Cones* depth image with holes.



(a)



(b)

Figure E.24: *Cones* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.25: *Dolls* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.26: *Dolls* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.27: *Dolls* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.28: *Dolls* texture image inpainted using ASC-JTDI.

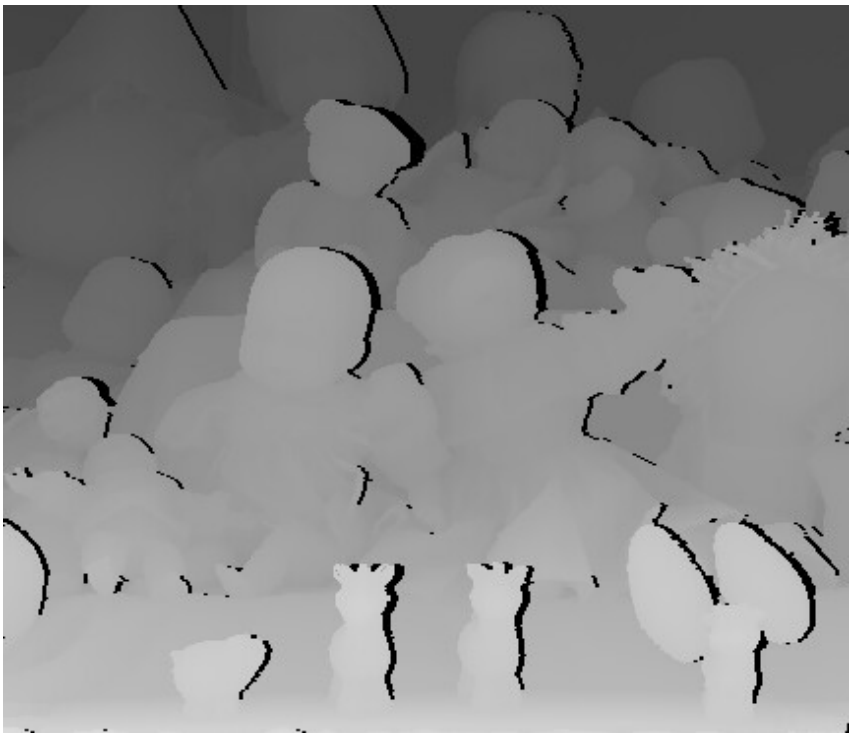
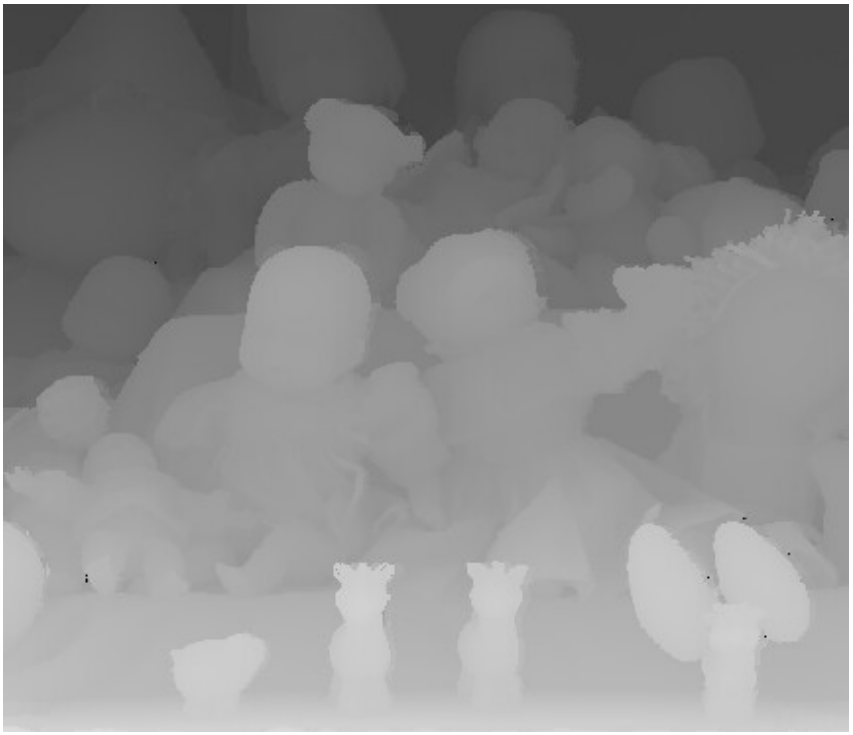
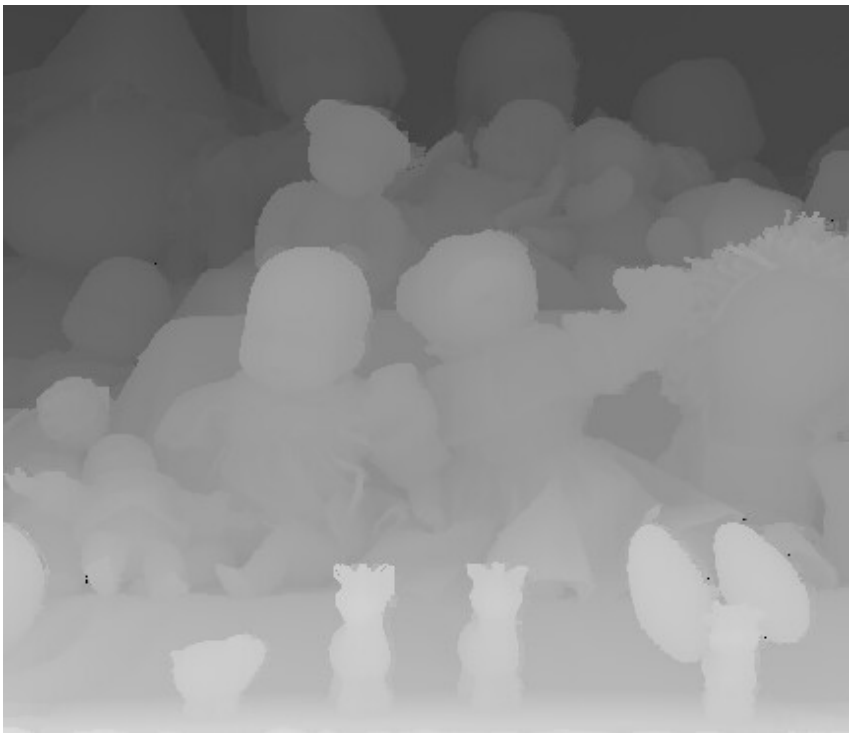


Figure E.29: *Dolls* depth image with holes.



(a)



(b)

Figure E.30: *Dolls* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.31: *Laundry* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.32: *Laundry* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.33: *Laundry* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.34: *Laundry* texture image inpainted using ASC-JTDI.

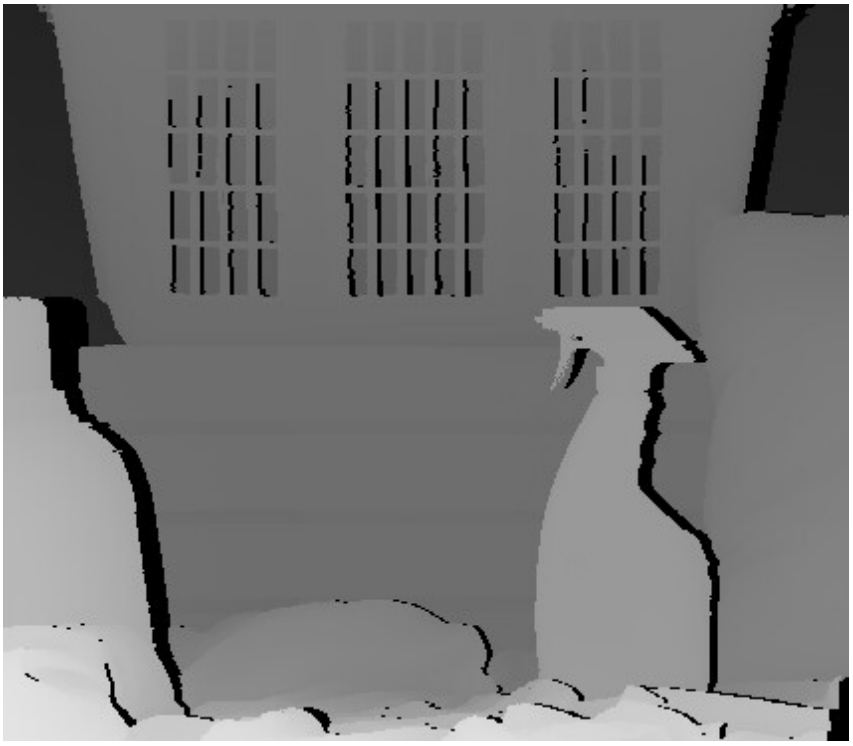
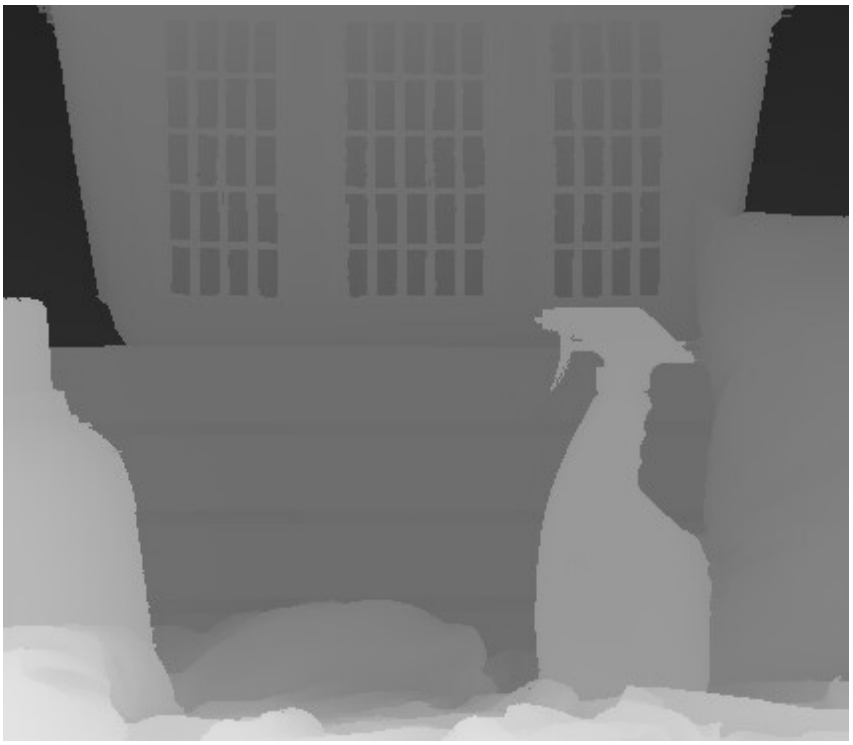
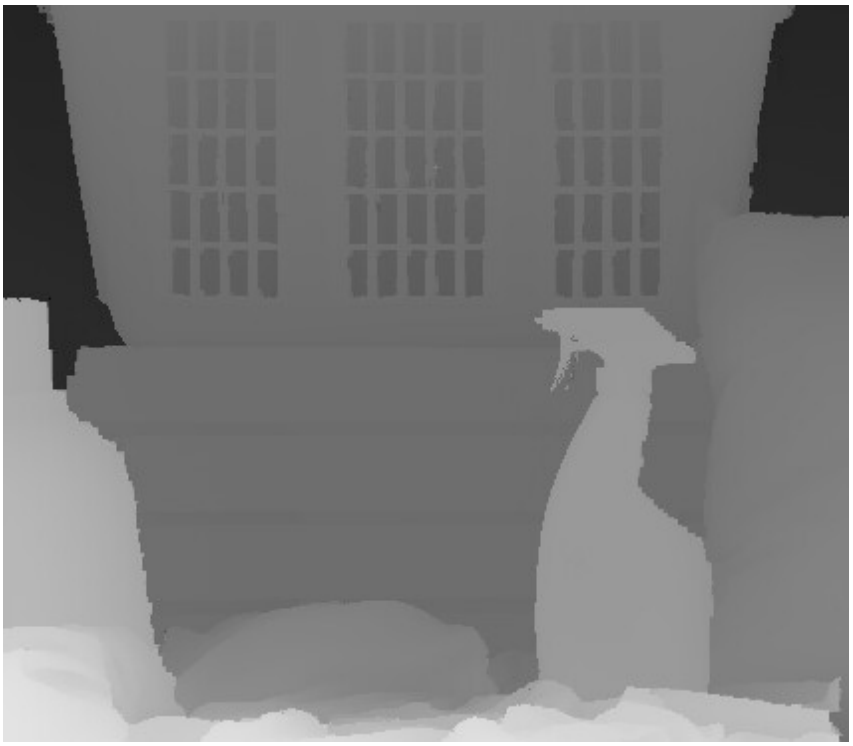


Figure E.35: *Laundry* depth image with holes.



(a)



(b)

Figure E.36: *Laundry* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.37: *Midd1* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.38: *Midd1* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.39: *Midd1* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.40: *Midd1* texture image inpainted using ASC-JTDI.

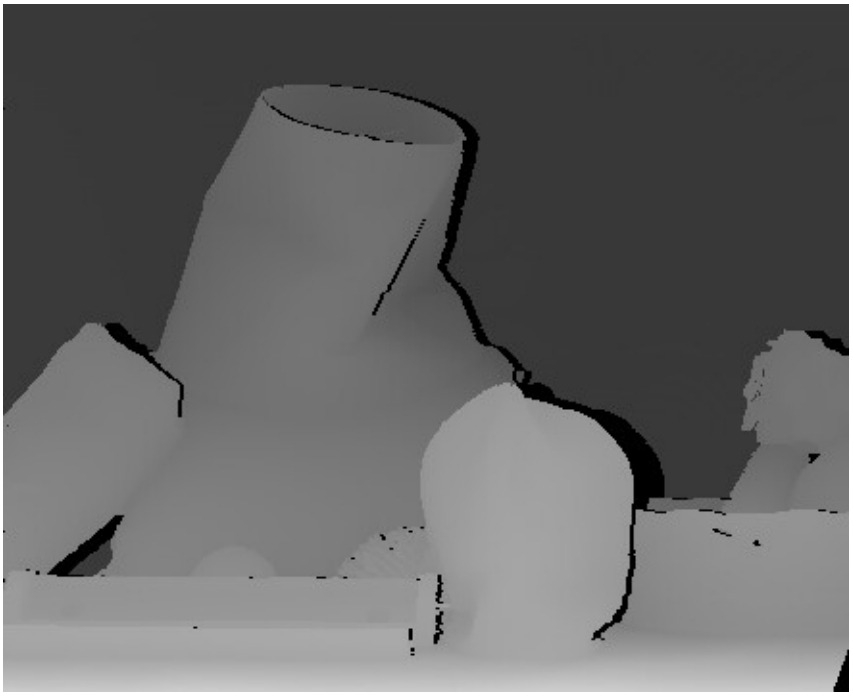
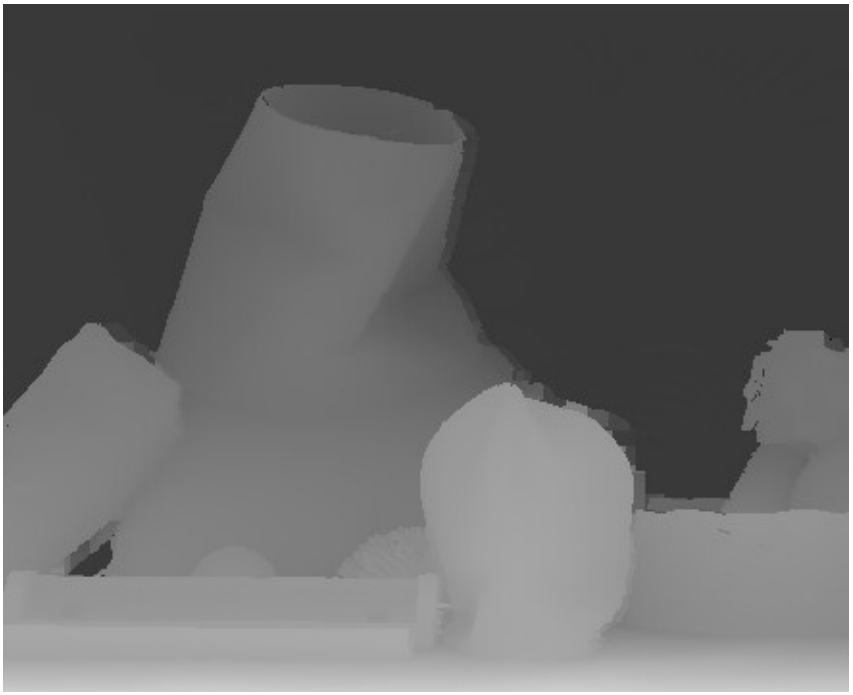
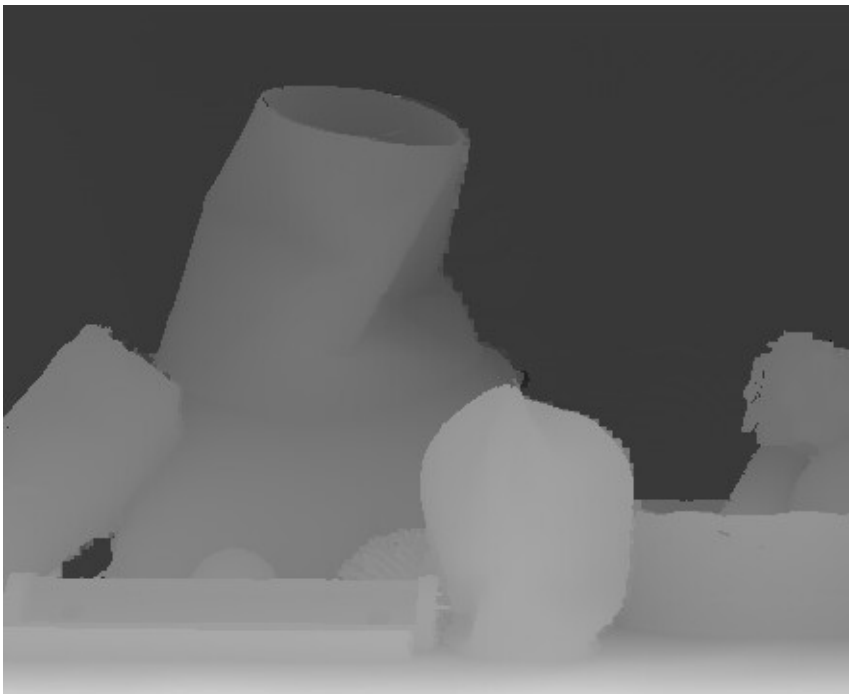


Figure E.41: *Midd1* depth image with holes.



(a)



(b)

Figure E.42: *Midd1* depth image inpainted using (a) Extrapolation and (b) JTDI.

(a)

(b)

Figure E.43: *Teddy* texture image with (a) holes and its corresponding (b) ground truth.

(a)

(b)

Figure E.44: *Teddy* texture image inpainted using (a) EBI and (b) DAI.

(a)

(b)

Figure E.45: *Teddy* texture image inpainted using (a) JTDI and (b) SC-JTDI.

Figure E.46: *Teddy* texture image inpainted using ASC-JTDI.

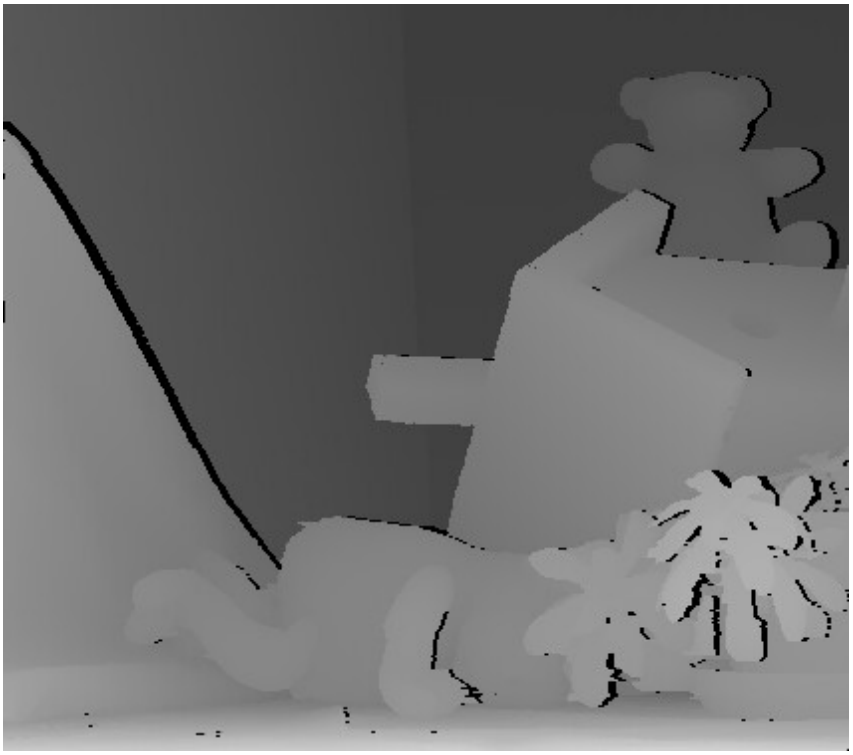
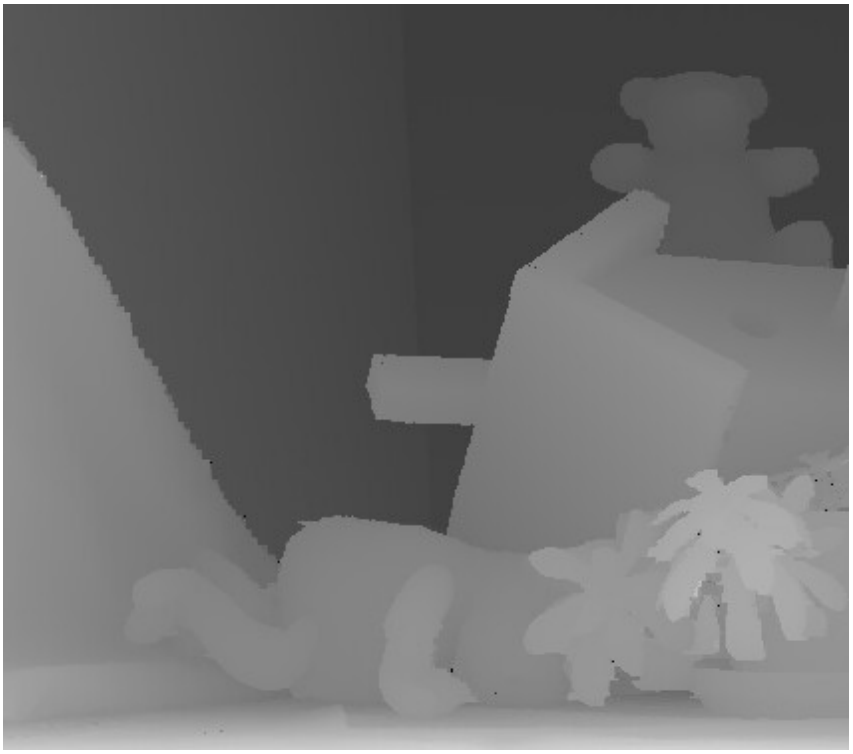
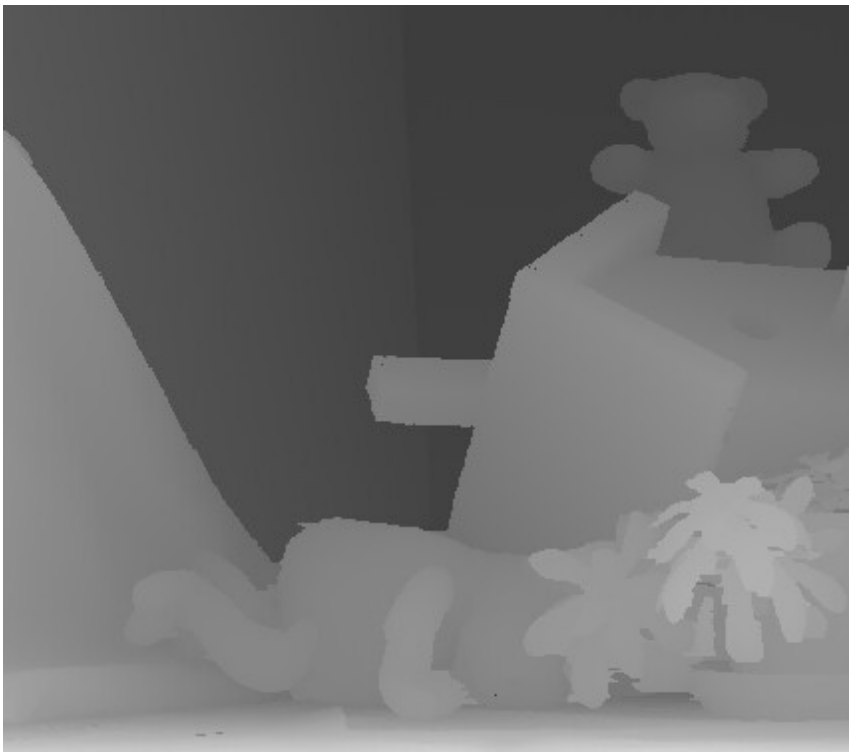


Figure E.47: *Teddy* depth image with holes.



(a)



(b)

Figure E.48: *Teddy* depth image inpainted using (a) Extrapolation and (b) JTDI.

Appendix F

Supplementary Literature

F.1 Log Polar Transform

The mathematical expression of mapping Cartesian coordinates $I(x, y)$ to the log-polar coordinates $LP(\rho, \theta)$is:

$$\rho = log_{base}\sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{F.1}$$

$$\theta = tan^{-1}\frac{y - y_c}{x - x_c} \tag{F.2}$$

Where (x, y) denotes the sampling pixel in the Cartesian coordinates and (x_c, y_c) is the centre pixel of transformation in the Cartesian coordinates. (ρ, θ) denotes the log-radius and the angular position in the log-polar coordinates and a natural logarithmic base.

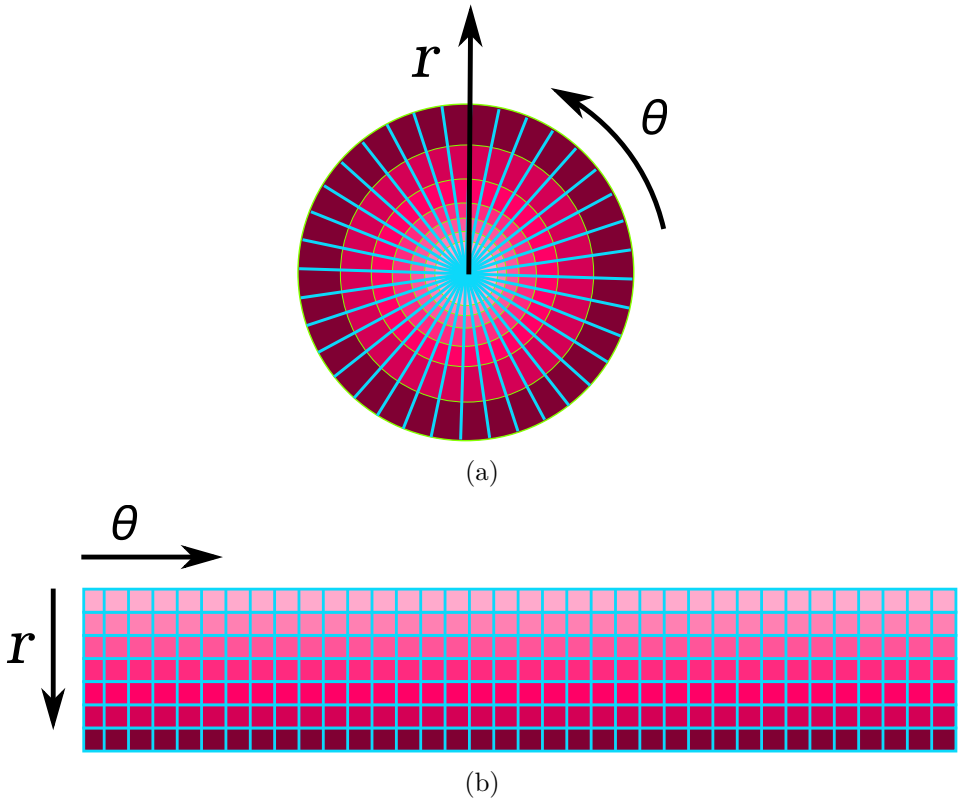


Figure F.1: LPT mapping: (a) LPT sampling in the Cartesian Coordinates, (b) the transformed result in the angular θ and log-radius r directions (Matungka, 2009).

Matungka (Matungka, 2009) explains the Log Polar Transform using the pictorial representation as shown in Figures, F.1 and F.2. Figure F.1 shows an example of the sampling point for image in the Cartesian coordinates and the transformed result.

As shown in Figure F.1(a), the distance between two consecutive sampling points in the radius direction increases exponentially from the centre to the furthest circumference. In the angular direction, for each radius, the circumference is sampled with the same number of samples. Hence, image pixels close to the centre are oversampled while image pixels further away from the centre are under-sampled or missed.

The advantage of using log-polar over the Cartesian coordinate representation is that any rotation and scale in the Cartesian coordinate representation is represented as shifting in the angular and the log-radius directions in the log-polar coordinates, respectively. Given $l(x', y')$ a scaled and rotated image of $f(x, y)$ with scale, rotation values a and ξ degrees, respectively, we have:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a\cos\xi & -a\sin\xi \\ a\sin\xi & a\cos\xi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{F.3}$$

$$x' = ax\cos\xi - ay\sin\xi,\ y' = ax\sin\xi + ay\cos\xi \tag{F.4}$$

In log-polar coordinate, $f(\rho, \theta) \rightarrow l(\rho', \theta')$,we have:

$$\rho' = log_{base}\sqrt{(ax\,cos\,\xi - ay\,sin\,\xi)^2 + (ax\,sin\,\xi + ay\,cos\,\xi)^2} \tag{F.5}$$

$$\rho' = log_{base}\sqrt{(ar\,cos\,\theta\,cos\,\xi - ar\,sin\,\theta\,sin\,\xi)^2 + (ar\,cos\,\theta\,sin\,\xi + ar\,sin\,\theta\,cos\,\xi)} \tag{F.6}$$

$$\rho' = log_{base}\sqrt{(ar\,cos(\theta+\xi))^2 + (ar\,sin(\theta+\xi))^2} = log_{base}\sqrt{a^2r^2} = \rho + log_{base}(a) \tag{F.7}$$

and

$$\theta' = tan^{-1}\left(\frac{y'}{x'}\right) = tan^{-1}\left(\frac{ax\,sin\,\xi + ay\,cos\,\xi}{ax\,cos\,\xi + ay\,sin\,\xi}\right) \tag{F.8}$$

$$\theta' = tan^{-1}\left(\frac{ar\,cos\,\theta\,sin\,\xi + ar\,sin\,\theta\,cos\,\xi}{ar\,cos\,\theta\,cos\,\xi - ar\,sin\,\theta\,sin\,\xi}\right) \tag{F.9}$$

$$\theta' = tan^{-1}\left(\frac{ar\,sin\,(\theta+\xi)}{ar\,cos\,(\theta+\xi)}\right) = \theta + \xi \tag{F.10}$$

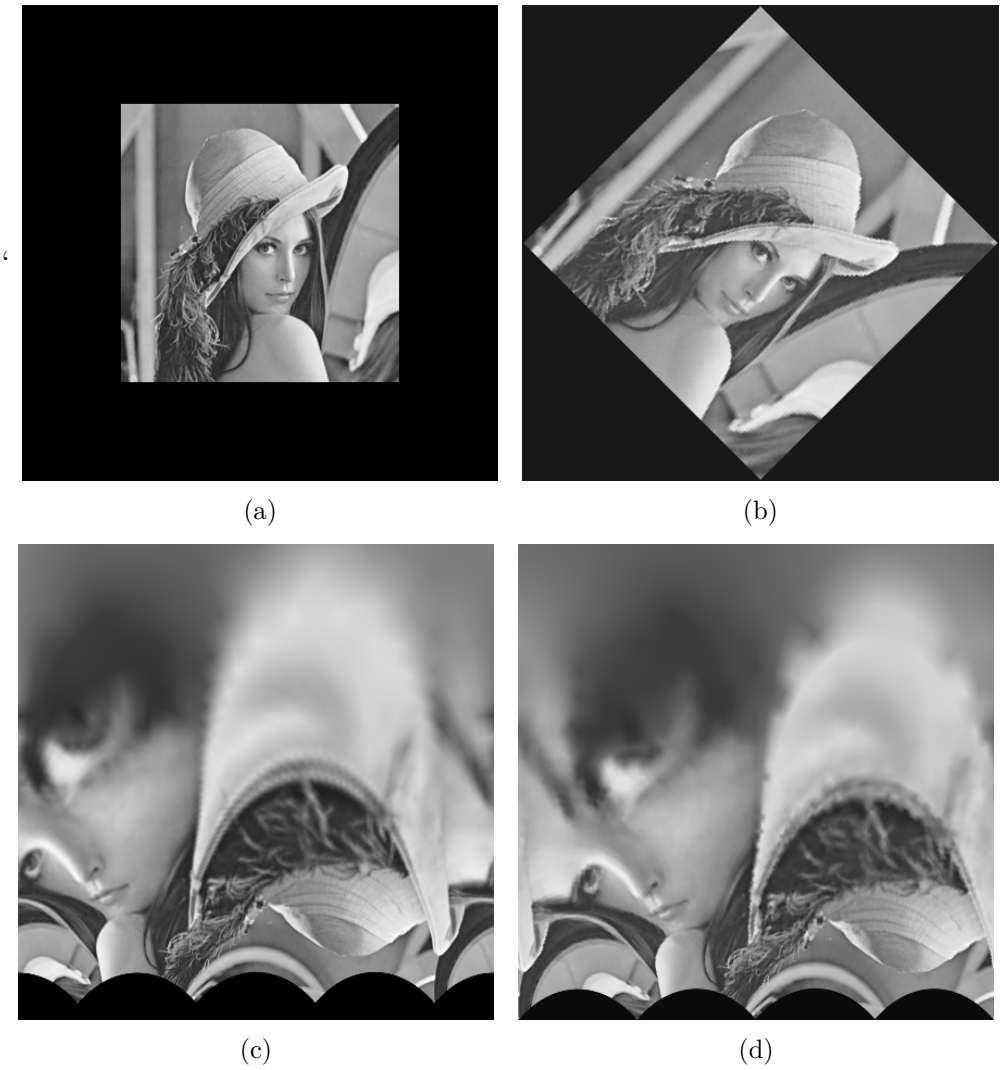


Figure F.2: (a) The Lena image, (b) the scaled and rotated image of (a), (c) the LPT image of (a), and (d) the LPT image of (b) (Matungka, 2009)

The advantage of using log-polar over the Cartesian coordinate representation is that any rotation and scale in the Cartesian coordinates is represented as shifting in the angular and the log-radius directions in the log-polar coordinates, respectively, as shown in Figure F.2. Figure F.2 (a) is the original image and Figure F.2 (b) is the scaled and rotated version of the original image. Figures, F.2 (c) and (d) are the LPT images of Figures, F.2(a) and (b), respectively.

The column of the log-polar coordinates represents the angular direction while the row represents the log-radius, thus rotation and scale in the Cartesian coordinates are represented as shifting in the log-polar coordinates.

F.2 Fourier Mellin Transform

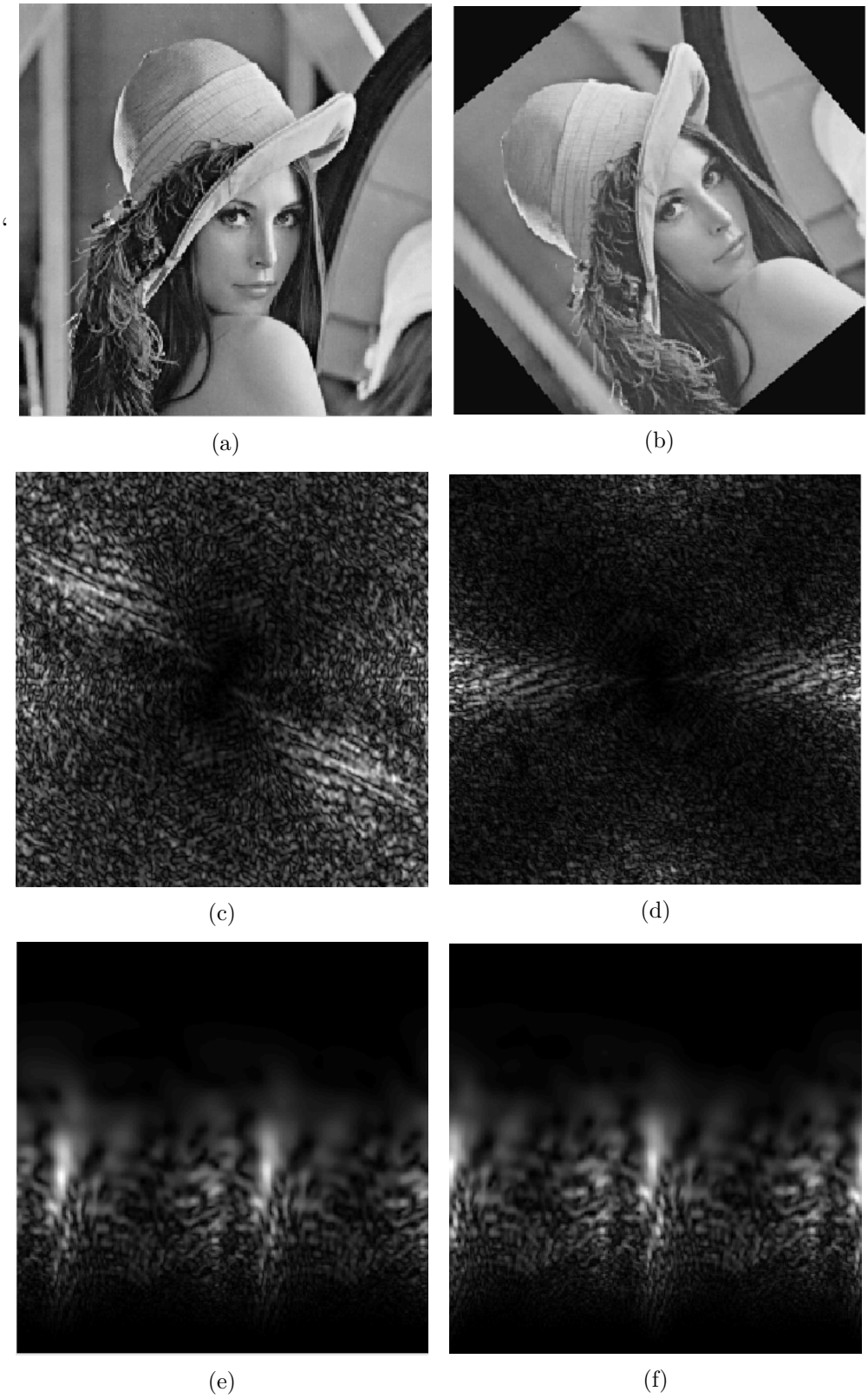
Fourier Mellin Transform is used to recover rotation and scale between two similar images (Bozek and Pivarciova, 2012; Sarvaiya et al., 2009). A worked example is used to show the various steps involved in determining the scale and rotation between two images. Following are the steps:

1. Load the two input images: an image 1 and image 2 which is scaled and rotated version of image 1. Figure F.3(a) shows an example Lena image of size 256×256 pixels, used as reference image. It is rotated to 40° and scaled by a factor 1.2 as shown in Figure F.3 (b).

2. Calculate 2D - FFT of both the images to attain two 2D arrays of FFT coefficient.

3. Perform convolution of magnitude spectrum of FFT coefficients. The outcome of this step is shown in Figure F.3 (c) and (d).

4. To calculate the scale and rotation between the images, the modified FFT arrays are transformed to log polar space (ρ, θ). The computed LPT images corresponding to Figure F.3 (c) and (d) are illustrated in Figure F.3 (e) and (f) respectively.

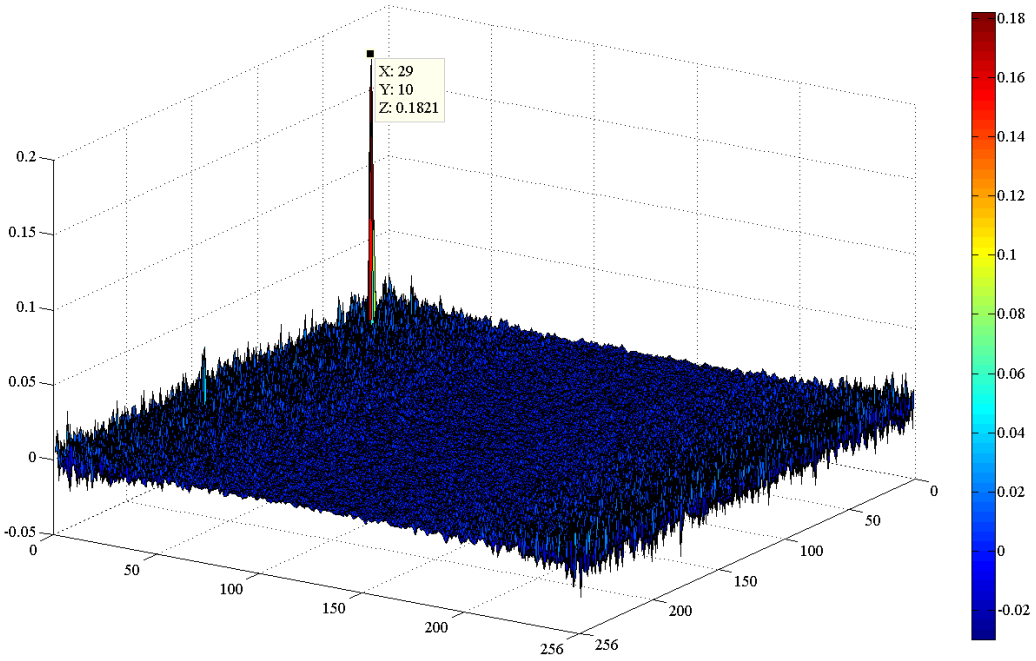


(a) (b)

(c) (d)

(e) (f)

Figure F.3: Lena (a) image 1, (b) image 2, (c) and (d) represent magnitude spectrum of (a) and (b), (e) and (f) corresponds to LPT of (c) and (d)



(a)

(b)

Figure F.4: (a) Cross power spectrum representing maximum magnitude peak $R_{peak}(x, y)$, (b) Final overlaid images.

5. Compute the cross power spectrum to determine the maximum magnitude peak R_{peak}. Figure F.4 (a) show the maximum magnitude peak R_{peak} of the cross power spectrum.

6. Based on the $(x,\ y)$ coordinates of R_{peak}, rotation angle is calculated as:

$$rotation\ angle = degrees\ per\ pixel\ \times (y-1) \tag{F.11}$$

where $degrees\ per\ pixel = 360°/size\ of\ image.$

The scale is computed as:

$$scale = \begin{cases} 1 & if\ x = 0 \\ (rho(-x+1) + rho(-x+2))/2 & x < 0 \\ 2/(rho(x+1) + rho(x+2)) & otherwise \end{cases} \tag{F.12}$$

For the calculation of the scale of the vector with the logarithmic layout between $\langle log10;\ log10^m \rangle$ is used as:

$$m = min([Ac - Cen(1)\ Cen(1) - 1\ Ar - Cen(2)\ Cen(2) - 1]); \tag{F.13}$$

$$rho = logspace(log_{10}(1),\ log_{10}(m),\ Nrho); \tag{F.14}$$

Where $Nrho$ is the number of points-lines of the transformed image, Ac is the number of columns of the input image, Ar is the number of rows of the input image, Cen is the center of the input image.

As shown in Figure F.4 (a), $R_{peak} = 0.1821$ is observed at $(x,\ y) = (10,\ 29)$. Using (F.11) - (F.14), the rotation angle and scale are computed as 39.35° and 1.19 respectively. Figure F.4 (b) shows the final image, after Image 2 is scaled and rotated using calculated values and overlaid over image 1.

From this example, this is evident that FMT helps in determining the rotation angle and scale values between similar images